Basic Programs for Investigating Relationships Between Output of Cohort Analysis and Independent Indices of Fish Stock Abundance

R. M. Korver

Department of Fisheries and Oceans
Resource Research Branch
240 Sparks Street
Ottawa, Ontario K1A 0E6

April 1983

Canadian Technical Report of Fisheries and Aquatic Sciences No. 1160

Canadian Technical Report of Fisheries and Aquatic Sciences

These reports contain scientific and technical information that represents an important contribution to existing knowledge but which for some reason may not be appropriate for primary scientific (i.e. *Journal*) publication. Technical Reports are directed primarily towards a worldwide audience and have an international distribution. No restriction is placed on subject matter and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries management, technology and development, ocean sciences, and aquatic environments relevant to Canada.

Technical Reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report will be abstracted in *Aquatic Sciences and Fisheries Abstracts* and will be indexed annually in the Department's index to scientific and technical publications.

Numbers 1-456 in this series were issued as Technical Reports of the Fisheries Research Board of Canada. Numbers 457-714 were issued as Department of the Environment, Fisheries and Marine Service, Research and Development Directorate Technical Reports. Numbers 715-924 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Technical Reports. The current series name was changed with report number 925.

Details on the availability of Technical Reports in hard copy may be obtained from the issuing establishment indicated on the front cover.

Rapport technique canadien des sciences halieutiques et aquatiques

Ces rapports contiennent des renseignements scientifiques et techniques qui constituent une contribution importante aux connaissances actuelles mais qui, pour une raison ou pour une autre, ne semblent pas appropriés pour la publication dans un journal scientifique. Il n'y a aucune restriction quant au sujet, de fait, la série reflète la vaste gamme des intérêts et des politiques du Ministère des Pêches et des Océans, notamment gestion des pêches, techniques et développement, sciences océaniques et environnements aquatiques, au Canada.

Les Rapports techniques peuvent être considérés comme des publications complètes. Le titre exact paraîtra au haut du résumé de chaque rapport, qui sera publié dans la revue *Aquatic Sciences and Fisheries Abstracts* et qui figurera dans l'index annuel des publications scientifiques et techniques du Ministère.

Les numéros 1-456 de cette série ont été publiés à titre de Rapports techniques de l'Office des recherches sur les pêcheries du Canada. Les numéros 457-714, à titre de Rapports techniques de la Direction générale de la recherche et du développement, Service des pêches et de la mer, ministère de l'Environnement. Les numéros 715-924 ont été publiés à titre de Rapports techniques du Service des pêches et de la mer, Ministère des Pêches et de l'Environnement. Le nom de la série a été modifié à partir du numéro 925.

La page couverture porte le nom de l'établissement auteur où l'on peut se procurer les rapports sous couverture cartonnée.

Canadian Technical Report of
Fisheries and Aquatic Sciences 1160

April 1983

BASIC PROGRAMS FOR INVESTIGATING RELATIONSHIPS BETWEEN OUTPUT OF COHORT ANALYSIS AND INDEPENDENT INDICES OF FISH STOCK ABUNDANCE

by

R. M. Korver

Department of Fisheries and Oceans
Resource Research Branch
240 Sparks Street
Ottawa, Ontario, K1A 0E6

©Minister of Supply and Services Canada 1983

Cat. No. Fs 97-6/1160E ISSN 0706-6457

Correct citation for this publication:

Korver, R. M. 1983. BASIC programs for investigating relationships between output of cohort analysis and independent indices of fish stock abundance. Can. Tech. Rep. Fish. Aquat. Sci. 1160:iv + 103 p.

CONTENTS

Abstract/Résumé iv

Introduction 1

1. General Instructions 1

2. Utility Programs 3

3. Data Entry Program 4

4. Cohort Analysis Program 11

5. Plotting and Line-fitting Program 25

6. Residuals Program 38

Conclusion 40

Acknowledgement 40

References 40

Appendix 1: System Requirements 41

Appendix 2: Listing for AUTOEXEC.BAT 41

Appendix 3: Listing for SYSMENU.BAS 41

Appendix 4: Listing for ENTRY.BAS 42

Appendix 5: Listing for COHORT.BAS 52

Appendix 6: Listing for COHORTC.BAS 67

Appendix 7: Listing for PREPLOT.BAS 82

Appendix 8: Listing for PLOT.BAS 83

Appendix 9: Contents of files containing F values 93

Appendix 10: Listing for RESIDS.BAS 94

Appendix 11: Printouts of the example data 99

ABSTRACT

Korver, R. M. 1983. BASIC programs for investigating relationships between output of cohort analysis and independent indices of fish stock abundance. Can. Tech. Rep. Fish. Aquat. Sci.1160:iv + 103p.

A package of computer programs has been developed to explore how changes in input parameters to cohort analysis affect the relationship of subsequent output with independent indices of fish stock abundnace. The package was written in BASIC on the IB" Personal Computer and contains programs for data entry, cohort analysis, plotting and line-fitting, and display of residuals. There is considerable flexibility in the selection and order of execution of the programs due to the interactive menu-driven format of the package. The report includes a description of calculation algorithms, a series of tutorial examples, and a complete listing of programs.

Key words: cohort analysis, indices of abundance, fisheries stock assessment, regression.

RÉSUMÉ

Korver, R. M. 1983. BASIC programs for investigating relationships between output of cohort analysis and independent indices of fish stock abundance. Can. Tech. Rep. Fish. Aquat. Sci.1160:iv + 103p.

Nous avons mis au point des programmes informatisés dans le but d'explorer comment les changements dans les entrées pour l'analyse par cohorte influent sur la relation entre les sorties ultérieures et les indices indépendants de l'abondance des stocks de poissons. Ces programmes, rédigés en langage BASIC à l'aide d'un IBM Personal Computer, couvrent l'entrée des données, l'analyse par cohorte, le traçage et l'ajustement de courbes et l'inspection des valeurs résiduelles. Une série de "menus" et un format interactif donnent à ces programmes une grande flexibilité quant au choix des programmes à exécuter et à l'ordre d'exécution. Le rapport comprend une description des algorithmes pour le calcul, une série d'examples pratiques et un listage complet des programmes.

Mots-clés: analyse par cohorte, indices d'abondance, évaluation des stocks de poissons, régression.

INTRODUCTION

A flexible interactive package of BASIC programs for cohort analysis (by the method of Pope, 1972) has been developed. With this package, input data for the analysis may be entered, the analysis itself may be performed, and the relationship of output to independent indices of fish stock abundance may be explored by regression analysis.

Clearly, changes in input parameters will affect the regression relationship. Adjustment of these parameters (especially terminal fishing mortalities) to achieve a "good" relationship is commonly referred to as the "fine-tuning" or calibration of cohort analysis. This process is important for accurate results in recent years of the analysis. However, there is variation and some subjectivity in the specific techniques used for tuning, and even in the definition of a "good" relationship.

The programs were designed to handle this variation but are by no means to be taken as a definitive statement on the techniques used in tuning a cohort analysis. In fact, one of the one of the primary reasons for the development of this package was to use it as a research tool for the investigation and improvement of present tuning methods. This report will stress description of the package rather than discussion of techniques. It is the user's responsibility to select a suitable approach for a given application.

The tuning process could become quite expensive on mainframe computing systems as a result of its iterative trial-and-error nature. An advantage of a microcomputer for this type of work is its essentially cost-free operation once the basic unit has been purchased. Program development costs are also much lower, being primarily limited to the cost of a programmer.

1. GENERAL INFORMATION

The programs were developed on an IBM Personal Computer, and use many features that are specific to the machine (for example, the user-definable "F-keys" described below.) Modifications will be required to implement this package on BASIC systems other than the IBM PC. The many versions of BASIC available on the market make it nearly impossible to provide generalized programs and documentation suitable for even the more popular hardware/software configurations. Hence, to avoid confusion, the programs will be described as they apply to the IBM PC. Appendix 1 gives a description of the system configuration used to develop the package and some alternative arrangements for running the programs without modification. For details on the IBM system not discussed in this report, the user is referred to the documentation supplied with the machine.

The programs are for the most part menu-driven and efforts have been made to keep them as self-documenting as possible. Thus, a large part of the report on the operation of the programs consists of tutorial-type example runs. Throughout, samples from the screen have been enclosed in boxes. Any information entered by the user has a solid underline.

To select an item from a menu, the "F-keys" on the left-hand side of the keyboard are used. These are numbered F1 through F10 and correspond to items in the menu displayed on the screen. Some options require that the Ctrl key be pressed simultaneously.

Figure 1 presents a generalized flow chart of the programs in the package. A synopsis of their function is provided in Table 1. Program listings are found in the Appendices.

STARTING THE PROGRAMS

Insert the diskette containing the programs into drive A (left side) and a diskette to be used for data storage in drive B (right side).

Turn the machine on, or if the machine is already on, press the **Ctrl**, **Alt**, and **Del** keys simultaneously (reset system). The program AUTOEXEC.BAT will execute automatically and request the date (see Utility Programs and Appendix 2). This program will load the BASIC program SYSMENU.BAS (see Utility Programs and Appendix 3) where the user chooses to begin the data entry program (ENTRY.BAS) or the first component of the cohort analysis program (COHORT.BAS). Alternatively, if one is in BASIC command level, then LOAD "ENTRY or LOAD "COHORT and RUN may be entered.

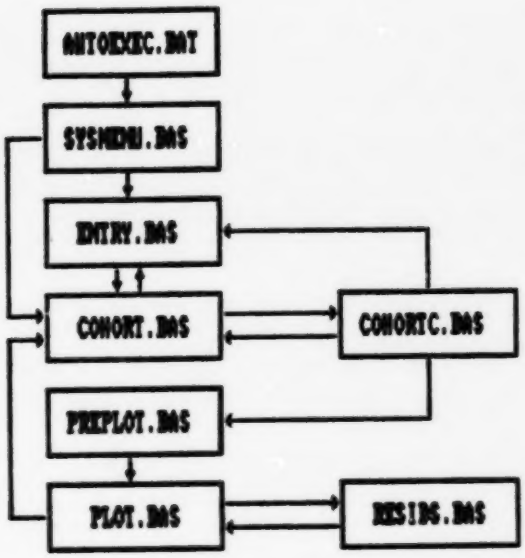


Figure 1. Generalized flow diagram showing the possible routes between the programs in the cohort analysis package.

Table 1. Synopsis of program functions.

Program Name	Function
AUTOEXEC.BAT	Executes automatically after system reset. User enters date. Loads BASIC and starts SYSMENU.BAS.
SYSMENU.BAS	User enters choice between beginning execution of data entry program (ENTRY.BAS) or cohort analysis programs (beginning with COHORT.BAS).
ENTRY.BAS	Data entry program. User may enter, load, edit, print, and store matrix or vector data used as input to cohort analysis programs.
COHORT.BAS	Contains input and output menus for cohort analysis. Input menu allows user to input data stored in ENTRY.BAS and to enter or change certain other data required for cohort analysis. The output menu allows for any input and calculated data currently defined to be displayed on the screen, printed, or output to diskette.
COHORTC.BAS	Contains calculation and output menus for cohort analysis. Interfaces closely with COHORT.BAS; all variables defined in COHORT.BAS remain defined here, and vice versa. The actual cohort analysis is performed by invoking the calculation menu. Output menu is nearly identical to that in COHORT.BAS, except that individual items from the calculation menu are routed directly to the output menu. In addition, these items may be sent to PLOT.BAS.
PREPLOT.BAS	Prepares selected calculation menu item for PLOT.BAS. Erases all previous calculated data (to clear memory for PLOT.BAS).
PLOT.BAS	Plotting and fitting program. Uses independent abundance index input in COHORT.BAS as independent variable, and calculation menu item selected in COHORTC.BAS as dependent variable. Plots graph(s) of data. Data may then be fitted to various regression models (linear, linear forced through origin, geometric mean, quadratic). Can return to COHORT.BAS for more calculations.
RESIDS.BAS	Calculates, plots, and prints residuals for model selected in PLOT.BAS.

2. UTILITY PROGRAMS

AUTOEXEC.BAT (listing in Appendix 2)

This program is executed automatically whenever the Disk Operating System (DOS) is loaded. The user enters the date (used for dating any diskette files and printouts created) in the format m-d-y or m/d/y. The program then loads advanced BASIC (BASICA.COM) and loads and executes SYSMENU.BAS.

```
A>date
Current date is 00-00-80
Enter new date: 12-17-82
A>basica SYSMENU
```

SYSMENU.BAS (listing in Appendix 3)

The user is given a choice between beginning execution of the data entry program (ENTRY.BAS) or the cohort analysis program (COHORT.BAS). This program is executed automatically after AUTOEXEC.BAT.

```
>>>   COHORT   SYSTEM   <<<

SELECT AN OPTION :

1 - ENTRY  Pgm. :  To enter and store matrices as files.

2 - COHORT Pgm. :  To start the cohort computations

Enter a NUMBER
1
```

MISCELLANEOUS

A number of other utilities having the filename extension .COM must reside on the same diskette with the cohort analysis package. The core DOS programs (boot record, IBMBIO.COM, IBMDOS.COM, COMMAND.COM), advanced BASIC (BASICA.COM), and the program DATE.COM are all required by the package. The user may add other DOS utilities as desired to fill the remaining portion of the program diskette.

3. DATA ENTRY PROGRAM

PROGRAM NAME

ENTRY.BAS (listing in Appendix 4)

PURPOSE

To enter, verify, print, and store vector or matrix data to be used as input for cohort analysis.

GENERAL REMARKS

The annotated tutorial run below gives the detailed instructions for the use of this program.

It is recommended that the user initially follows the order of the menu, however, the order of execution is quite flexible. NOTE: If a vector or matrix is present in memory, its contents will be deleted if option F1 or F5 is selected (see below). For F5, the contents are replaced as each year of data is entered.

The program uses modified cursor editing algorithms in the Data Entry (F5) and Edit (F6) options (see tutorial run for instructions). This allows for considerable flexibility in entry, verification, and correction of data.

The data is organized to have ages along the row dimension of the vector/matrix and years along the column dimension. If the data is a vector whose elements correspond to ages, then it will be entered, displayed, and printed as a column. The year specified by F3 will be printed at the top of the column. If the data is a vector whose elements correspond to years, then it will be entered, displayed, and printed as a row. NO age will be shown.

To use the numeric key pad on the right-hand side of the keyboard, the **Num Lock** key is pressed once. NOTE: Remember to press it again if the cursor movement, **Pg Up** or **Pg Dn** key functions are required (see tutorial run for the use of these keys).

The maximum dimensions for a matrix are 20 ages by 20 years.

EXAMPLE TUTORIAL RUN

The example tutorial runs in this report use printed copies of the screen while the programs are running. Data for cod in NAFO area 4VsW (Gray, 1979) are used as an example. Execution of the data entry program begins with display of the menu below.

```
                DATA ENTRY PROGRAM MENU

Enter one choice by pressing the appropriately numbered F-key.

F1.  DIMENSIONS OF DATA (number of age groups, number of years)
F2.  TITLE (optional - default is blank)
F3.  FIRST YEAR OF DATA (optional - default is 1)
F4.  FIRST AGE GROUP (optional - default is 1)
F5.  DATA ENTRY
F6.  EDIT
F7.  DISKETTE OUTPUT
F8.  DISKETTE INPUT
F9.  HARDCOPY (NOTE: Filename will be printed if F7 or F8 immediately precede)
F10. EXPAND MATRIX (adds years to matrix presently in memory)
CTRL F1. START COHORT ANALYSIS
CTRL F2. END
```

F1. Dimensions of data

Self-explanatory.

```
Entry of dimensions. Maximum 20 x 20 matrix.
NOTE: If data is a vector then enter '1' for the dimension
not to be used.
Enter number of years:  13
Enter number of age groups:  16
```

F2. Title

The title has two components: the type of data being entered (eg. catch at age; F, last year), and the identity of the stock in question (plus any other relevant identification).

```
Type of data is:
Change? (Y/N)  y
Enter type of data (<20 chars. - eg. Catch at age, Weight at age):
--------------------
Catch at age

Identity of stock is:
Change? (Y/N)  y
Enter identity of stock (<=45 chars.):
---------------------------------------------
4VsW Cod - Adjusted Data
```

F3. First year of data

```
Current first year is:  1
Change? (y/n)  y
Enter first year of data:  1966
```

F4. First age group

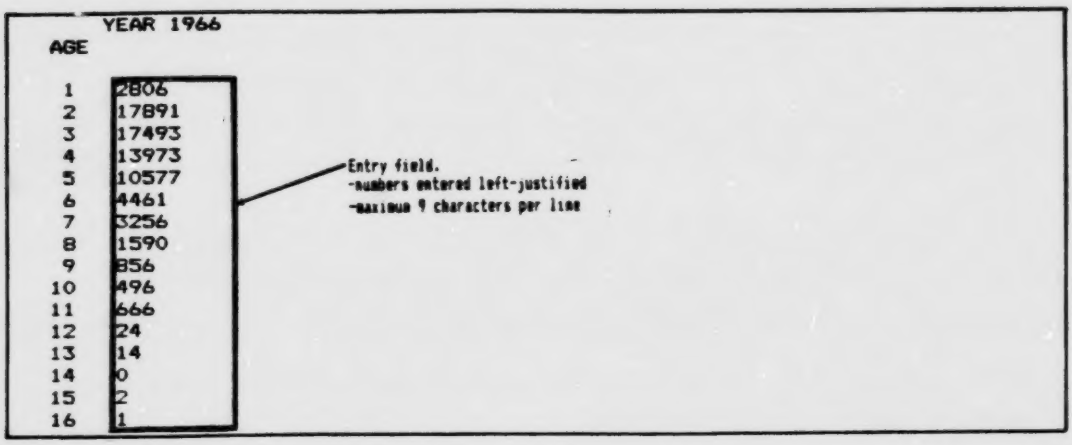
```
Current first age group is:  1
Change? (y/n)  n
```

F5. Data entry

The following explanation appears when this option is selected. The machine will emit a beep if an invalid character is entered (i.e. NOT the digits 0-9, +, -, decimal point, ↑, ↓, ←, →, backspace, space, carriage return, or N) or if attempting to leave the boundaries of the entry area (9 characters wide, and in this example, 16 lines deep).

```
Data is entered by year, pressing return after each number entered.
The year and ages are printed across the top and down the left
side of the screen, respectively. The user enters the data in
the appropriate location. If a mistake is made in entering the
data, use the cursor movement keys to move to the error and
type over it. The backspace key and space bar may also be used.
Make sure to begin each entry in the first field on the left of
the dark portion of the screen. To enter the next year of data,
press the letter N at any time. Pressing N after the last year of
data has been entered will return the user to the menu. If the
user wishes to return to the menu without entering data for all
years (eg. if very bad entry errors are made), then press N
successively to advance the screen through the remaining years.
Only 9 digits including decimals and negative signs are allowed.

Press any key to begin data entry.
```

On a new screen, the data for 1966 are entered.



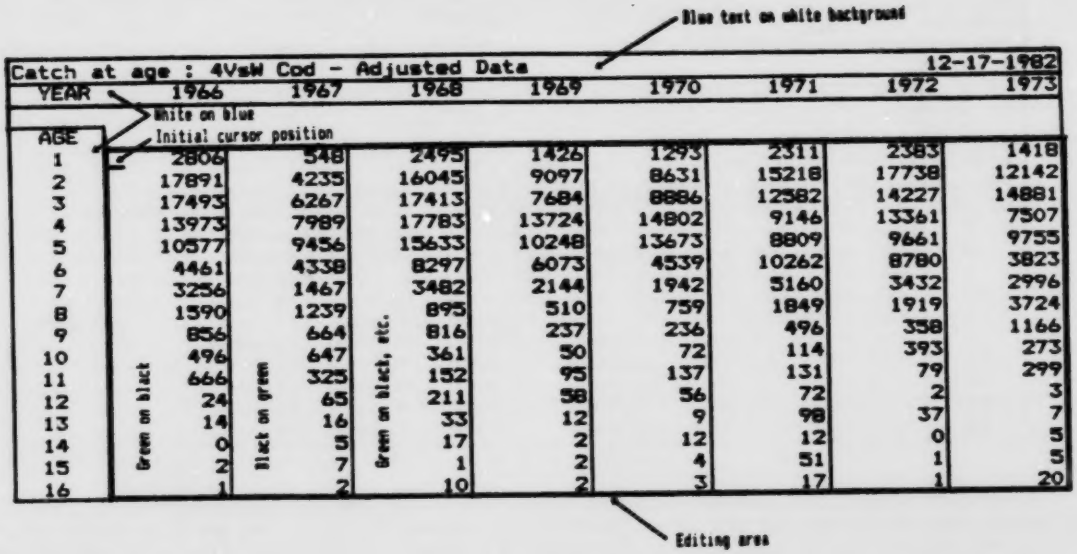
For 1967, press N. Continue data entry to 1978 (16 years). Pressing N after 1978 data is entered will return user to menu.

F6. Edit

The following instructions appear upon choosing this option.

```
Up to 8 years of data can be displayed on the screen for editing
at one time. To get the next 8 years press 'Pg Up'. To get the
previous 8 years, press 'Pg Dn'. To move the cursor to the desired
position for corrections, use ↑, ↓, →, ←. To move from a
given field (year) to the beginning of the next or previous one,
use 'Ctrl' + '→' or 'Ctrl' + '←' respectively.
To change a number, merely type over it. The backspace and return
keys, as well as the space bar may also be used. Only the matrix or
vector portion of the data may be changed in this sub-routine.
For corrections to title, first age, and first year, the appro-
priate sub-routine must be (re)invoked. To exit this routine,
press the letter 'X' at any time.

Enter desired number of decimal places (0-4):  0
```

The first 8 years of data are then displayed as below. The cursor appears as shown at the beginning of the first field (year 1966) for the first age group. The screen display is in colour for improved clarity. Colours are labelled (unlabelled areas are black). As in F5, a beep signal occurs for invalid characters and when one attempts to leave the boundaries of the editing area.



YEAR	1966	1967	1968	1969	1970	1971	1972	1973
AGE								
1	2806	548	2495	1426	1293	2311	2383	1418
2	17891	4235	16045	9097	8631	15218	17738	12142
3	17493	6267	17413	7684	8886	12582	14227	14881
4	13973	7989	17783	13724	14802	9146	13361	7507
5	10577	9456	15633	10248	13673	8809	9661	9755
6	4461	4338	8297	6073	4539	10262	8780	3823
7	3256	1467	3482	2144	1942	5160	3432	2996
8	1590	1239	895	510	759	1849	1919	3724
9	856	664	816	237	236	496	358	1166
10	496	647	361	50	72	114	393	273
11	666	325	152	95	137	131	79	299
12	24	65	211	58	56	72	2	3
13	14	16	33	12	9	98	37	7
14	0	5	17	2	12	12	0	5
15	2	7	1	2	4	51	1	5
16	1	2	10	2	3	17	1	20

By pressing 'Pg Up' the remaining data appears as below. If no changes have been made to the previous screen then it will immediately print the next screen. If some numbers were changed (the program checks this by setting a flag for the entry of a character, a space, or a backspace) then the cursor travels across the screen row by row, reading the edited data. The edited data are read from the screen as characters, converted to numeric, and then replace the old data.

Catch at age : 4VsW Cod - Adjusted Data 12-17-1982

YEAR	1974	1975	1976	1977	1978
AGE					
1	1482	1792	728	2	177
2	8451	9979	4061	24	153
3	12885	9485	3587	386	1004
4	9947	4341	3713	1073	3650
5	7130	4549	4818	1559	4621
6	2766	2594	2412	871	2441
7	944	2627	1426	501	768
8	1323	612	611	220	213
9	413	497	184	128	112
10	369	660	49	35	80
11	15	153	22	44	26
12	5	126	107	55	28
13	0	36	1	11	26
14	0	9	4	3	9
15	0	9	1	2	4
16	0	18	1	7	2

Problems may arise if the user has a matrix of large numbers (as in a catch matrix for example) and accidentally selects to edit it with more decimal places than were used in the Data Entry option. BASIC signals this condition by printing a percent sign (%) in front of any number that overflows its specified field. Below is an example created by requesting four decimal places for the 4VsW cod catch at age data.

Field overflow error: Please check the numbers for age 2 12-20-1982

YEAR	1966	1967	1968	1969	1970	1971	1972	1973
AGE								
1	2806.0000	548.0000	2495.0000	1426.0000	1293.0000	2311.0000	2383.0000	1418.0000
2	%17891.0000	4235.0000	%16045.0000	9097.0000	8631.0000	%15218.0000	%17738.0000	%1
2142.00	%17493.0000	6267.0000	%17413.0000	7684.0000	8886.0000	%12582.0000	%14227.0000	%1
4881.00	%13973.0000	7989.0000	%17783.0000	%13724.0000	%14802.0000	9146.0000	%13361.0000	
7507.00	%10577.0000	9456.0000	%15633.0000	%10248.0000	%13673.0000	8809.0000	9661.0000	97
55.0000	4461.0000	4338.0000	8297.0000	6073.0000	4539.0000	%10262.0000	8780.0000	3823.000
0 7	3256.0000	1467.0000	3482.0000	2144.0000	1942.0000	5160.0000	3432.0000	2996.0000
8	1590.0000	1239.0000	893.0000	510.0000	759.0000	1849.0000	1919.0000	3724.0000
9	856.0000	664.0000	816.0000	237.0000	236.0000	496.0000	358.0000	1166.0000
10	496.0000	647.0000	361.0000	50.0000	72.0000	114.0000	393.0000	273.0000
11	666.0000	325.0000	152.0000	95.0000	137.0000	131.0000	79.0000	299.0000
12	24.0000	65.0000	211.0000	58.0000	56.0000	72.0000	2.0000	3.0000
13	14.0000	16.0000	33.0000	12.0000	9.0000	98.0000	37.0000	7.0000
14	0.0000	5.0000	17.0000	2.0000	12.0000	12.0000	0.0000	5.0000
15	2.0000	7.0000	1.0000	2.0000	4.0000	51.0000	1.0000	5.0000
16	1.0000	2.0000	10.0000	2.0000	3.0000	17.0000	1.0000	20.0000

If this occurs, the user is advised to press "X" and run the Edit option again. If, however, changes have been made, and the screen is read, the program checks for the presence of the percent sign and displays the message seen at the top of the screen with a warning beep. The program then pauses for several seconds, and resumes reading at the beginning of the next age group. Thus no invalid fields are read. The user should check the values for the age group where the error occured as any changes made to the line will not be read.

F7. Diskette output

Data with 7 or less significant digits may be stored on diskette in single precision. This will not affect the precision of the calculations in the cohort analysis program; all calculations are performed in double precision (however, see note on precision under General Remarks in section 3). The single precision diskette output option is included to economize on storage space.

For each file created, the following are stored: type of data (F2), identity of stock (F2), storage precision ("d" or "s"), number of age groups, number of years, first age, first year, and the actual data matrix or vector.

```
Would you like single or double precision for diskette output? (S/D)  s

WARNING: Single precision output will only store 7 significant digits.
Any numbers with more than 7 significant digits will be rounded off.
Press 'D ' if you would rather have double precision, if not, press RETURN.
```

A carriage return was entered.

```
Rules for naming of files are as described in the BASIC reference
manual. If no filename extension is included then the extension
'.dat ' is added by default.

Enter filename:  4vswcod.cat
Insert data diskette into drive B.
Press any key to begin output to diskette.

Data has been stored on drive B as:   4vswcod.cat
Press any key to return to menu.
```

Below is another example of this option, which was done immediately after the above run. Double precision was chosen this time, and the same filename was entered.

```
Would you like single or double precision for diskette output? (S/D)  d

Rules for naming of files are as described in the BASIC reference
manual. If no filename extension is included then the extension
'.dat ' is added by default.

Enter filename:  4vswcod.cat
Insert data diskette into drive B.
Press any key to begin output to diskette.
```

When output to diskette is attempted, the following choices are displayed to prevent accidental loss of data:

```
File already exists. Would you like to:
  A. Enter a different filename?
  B. Rename the existing file without changing the name you have just entered?
  C. Write over the existing file?

Press A, B, or C on the keyboard.  c

Data has been stored on drive B as:  4vswcod.cat
Press any key to return to menu.
```

Appropriate prompts are displayed if option A or B is chosen.

Error trapping routines are also activated when a diskette is full (160K bytes) or when the maximum number of files per diskette is exceeded (64 files). The user is then requested to insert a new diskette.

F8. Diskette input

Self-explanatory.

```
This sub-routine reads data already stored on diskette so that
it may be verified and/or changed using the edit subroutine.
It may be rewritten to diskette with the output sub-routine.

Enter filename (including extension) of data:  4vswcod.cat
Insert data diskette into drive B.
Strike any key when ready.

Data was stored in double precision.
The file has been loaded. Press any key to return to menu.
```

If the file cannot be found (if one entered "4vswcod.car", for example) the following message will appear.

```
File not found. Probable cause: wrong diskette and/or filename.
Re-enter filename:  4vswcod.cat
```

F9. Hardcopy

This option outputs the data on the printer in a format comparable to that of the Edit option. The field width, however, is variable in order to maximize the number of years that are printed on the 132-column page width. If there are more years than can be fit on one page width, the remaining years are printed in one or more blocks (as required) below. If the user has just run the diskette output (F8) or input (F9) options, then the name of the diskette filename will be printed above the data for reference purposes.

Only the number of decimal places are required as input:

```
Enter desired number of decimal places (0-4):  0
```

The following printout results.

Name of diskette file is: 4vswcod.cat

Catch at age : 4VsW Cod - Adjusted Data 12-17-1982

YEAR	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978
AGE													
1	2806	548	2495	1426	1293	2311	2363	1418	1482	1792	728	2	177
2	17891	4235	16045	9097	8631	15218	17738	12142	8451	9979	4061	24	153
3	17493	6267	17413	7684	8886	12582	14227	14881	12885	9485	3587	386	1004
4	13973	7989	17783	13724	14802	9146	13361	7507	9947	4341	3713	1073	3650
5	10577	9456	15633	10248	13673	8809	9661	9755	7130	4549	4818	1559	4621
6	4461	4338	8297	6073	4539	10262	8780	3823	2766	2594	2412	871	2441
7	3256	1467	3482	2144	1942	5160	3432	2996	944	2627	1426	501	768
8	1590	1239	895	510	759	1849	1919	3724	1323	612	611	220	213
9	856	664	816	237	236	496	358	1166	413	497	184	128	112
10	496	647	361	50	72	114	393	273	369	660	49	35	80
11	666	325	152	95	137	131	79	299	15	153	22	44	26
12	24	65	211	58	56	72	2	3	5	126	107	55	28
13	14	16	33	12	9	98	37	7	0	36	1	11	26
14	0	5	17	2	12	12	0	5	0	9	4	3	9
15	2	7	1	2	4	51	1	5	0	9	1	2	4
16	1	2	10	2	3	17	1	20	0	18	1	7	2

F10. Expand matrix

This option is useful for updating data as new assessments becomes available. The data entry program is first used to create a file with the additional data on it. The matrix to be expanded is then loaded (F8), and this option then is selected with the following result.

```
This option appends years previously stored on diskette to the matrix
presently residing in memory.
NOTES: 1. Data to be added must have same number of age groups as present data.
       2. The expanded matrix must not have more than 20 years of data and
          will replace the present matrix.
       3. The present title is not changed.

Enter filename (including extension) of data to be appended:  4vswcod.add
Insert data diskette into drive B.
Strike any key when ready.

Expanded matrix has been created.
Press any key to return to menu.
```

The file called "4vswcod.add" contained catch data for 1979-80. The expanded matrix may now be edited, printed, and/or stored on diskette.

It is possible to delete years (or age groups) from the cohort analysis by selecting the desired interval(s) in the catch matrix input of COHORT.BAS. The interval(s) chosen there will be applied to all other data used in the analysis.

CTRL F1. Start Cohort Analysis

This option loads and begins execution of the main cohort analysis program and deletes the data entry program from memory.

CTRL F2. End

This option ends the program and return the user to BASIC command level. It also sets the following six F-keys (partially identical to those set when entering BASIC from DOS).

F1. LIST
F2. RUN← (RUN with a carriage return character)
F3. LOAD"
F4. SAVE"
F5. CONT←
F6. EDIT

4. COHORT ANALYSIS PROGRAM

PROGRAM NAME

COHORT.BAS, COHORTC.BAS (listings in Appendix 5 and 6)

PURPOSE

This program performs cohort analysis as described in Pope (1972). Several features that are useful in "tuning" or calibrating the analysis are included.

CALCULATION ALGORITHM

Much of the algorithm for the computations has been taken from Rivard (1982). His notation is preserved as much as possible.

Subscript Notation

- i : a given age (taken here in years)
- t : a given point in time (taken here as the beginning of a year)
- $\imath$: the age interval i, i+1
- τ : the time interval t, t+1
- t_0 : first year
- t_f : final year
- b : age of youngest age-group
- m : age of oldest age-group
- μ : the age interval m, m+1
- τ_f : the time interval t_f, t_f+1

Inputs

The possible inputs required for the calculations consist of:

$C_{\imath,\tau}$: catch at age in numbers ($i = b,\ldots m$; $t = t_0,\ldots t_f$); required

$W_{i+.5,t+.5}$: mid-year estimates of weight at age expressed kilograms ($i = b,\ldots,m$; $t = t_0,\ldots,t_f$); optional

$F_{\imath,\tau_f}$: instantaneous rate of fishing mortality for each age-group in the final year ($i = b,\ldots,m$); required

$F_{\mu,\tau}$: instantaneous rate of fishing mortality for the oldest age-group in each year ($t = t_0,\ldots,t_f$); optional

M : instantaneous rate of natural mortality (assumed constant for all ages and years); required, but default is 0.2

$r_{\imath,\tau_f}$: partial recruitment for each age-group ($i = b,\ldots,m$); optionally used to generate $F_{\imath,\tau_f}$

F_{τ_f} : fully recruited instantaneous fishing mortality for the final year; optionally used to generate $F_{\imath,\tau_f}$

The last two inputs can generate $F_{\imath,\tau_f}$ by using

$$F_{\imath,\tau_f} = r_{\imath,\tau_f} F_{\tau_f}. \qquad [3.1]$$

Details on the use of this option are found in the tutorial example run below.

Calculations

The calculations are performed as follows:

Population numbers (beginning of year), $N_{i,t}$:

Fishing is assumed to be incomplete for the oldest age-group. Using $i = b,\ldots,m-1$ and $t = t_f$, the $N_{i,t}$ for the final year are calculated as

$$N_{i,t_f} = C_{i,\tau_i} Z_{i,\tau_i} / [F_{i,\tau_i}(1-e^{-Z_{i,\tau_i}})] \quad , \qquad [3.2]$$

where $Z_{i,\tau_i} = F_{i,\tau_i} + M$ is the total mortality.

The same equation is used to compute population numbers for the oldest age group ($i=m$) in each year ($t = t_0,\ldots,t_f$).

The remaining numbers ($i = b,\ldots,m-1$; $t = t_0,\ldots,t_f-1$) are calculated using

$$N_{i,t} = C_{i,\tau} e^{M/2} + N_{i+1,t+1} e^{M} \quad . \qquad [3.3]$$

Population numbers (midyear), $N_{i+.5,t+.5}$:

$$N_{i+.5,t+.5} = N_{i,t} e^{-(Z_{i,\tau}/2)} \qquad i = b,\ldots m;\ t = t_0,\ldots,t_f. \qquad [3.4]$$

Population biomass (midyear), $B_{i+.5,t+.5}$

$$B_{i+.5,t+.5} = W_{i+.5,t+.5} N_{i+.5,t+.5} \qquad i = b,\ldots,m;\ t = t_0,\ldots,t_f. \qquad [3.5]$$

Population biomass (fishable), $BPR_{i+.5,t+.5}$

$$BPR_{i+.5,t+.5} = W_{i+.5,t+.5} N_{i+.5,t+.5} r_i \qquad i = b,\ldots,m;\ t = t_0,\ldots,t_f. \qquad [3.6]$$

The partial recruitment vector, r_i, is usually calculated from [3.13] or [3.14].

Age-specific fishing mortalities, $F_{i,\tau}$:

$$F_{i,\tau} = \ln (N_{i,t}/N_{i+1,t+1}) - M \qquad i = b,\ldots,m-1;\ t = t_0,\ldots, t_f-1. \qquad [3.7]$$

The program sets $F_{i,\tau}$ to zero if $N_{i+1,t+1} = 0$.

Totals for $N_{i,t}$; $N_{i+.5,t+.5}$; $B_{i+.5,t+.5}$:

Totals over ages for beginning of year population numbers, midyear population numbers, and midyear population biomass are computed in similar fashion. The general summing equation (with $t=t_0,\ldots,t_f$) is

$$A_{.,t} = \sum_{i=k}^{l} A_{i,t} \quad . \qquad [3.8]$$

where $i = k,\ldots,l$ is any selected age interval within the interval $i = b,\ldots,m$. For total fishable biomass, all age groups ($i = b,\ldots,m$) are used.

Mean fishing mortalities:

Weighted or unweighted mean annual rates of fishing mortality may be calculated. Weighted means are calculated as

$$\bar{F}_\tau = \left(\sum_{i=k}^{l} N_{i,t} F_{i,\tau}\right) / \left(\sum_{i=k}^{l} N_{i,t}\right) \quad . \qquad [3.9]$$

with $k,\ldots,l$ as above. Unweighted means are computed as

$$\bar{F}_\tau = \left(\sum_{i=k}^{l} F_{i,\tau}\right) / (l-k+1) \quad . \qquad [3.10]$$

Overall fishing mortalities:

This alternative method of summarizing fishing mortality is calculated from the equation

$$\bar{F}_\tau = \ln \left(\sum_{i=k}^{l-1} N_{i,t} / \sum_{i=k}^{l-1} N_{(i+1),(t+1)} \right) - M \quad . \qquad [3.11]$$

Partial recruitment (PR):

The basic equation used to find the relative catchability at age, or partial recruitment, is

$$r_{l,\tau} = F_{l,\tau} / F\max_\tau \quad , \qquad [3.12]$$

where $F\max_\tau$ is the maximum value of $F_{l,\tau}$ observed during any given period, τ. The matrix of $r_{l,\tau}$ may be averaged over years using an arithmetic or geometric mean. The resulting age-specific vector is restandardized. For the arithmetic mean, the equation is

$$r_l = \left(\sum_{t=p}^{q} r_{l,\tau} \right) / [(q-p+1)\ \mathrm{rmax}] \quad , \qquad [3.13]$$

where $t=p,\ldots,q$ is any selected interval of years within $t=t_0,\ldots,t_f$; rmax is the maximum value of $\left(\sum_{t=p}^{q} r_{l,\tau} \right)/(q-p+1)$.

The standardized geometric mean vector is found by using

$$r_l = \left(\prod_{t=p}^{q} r_{l,\tau} \right)^{1/(q-p+1)} / \mathrm{rmax}$$

$$= \exp\left[\left(\sum_{t=p}^{q} \ln r_{l,\tau} \right)/(q-p+1) \right] / \mathrm{rmax} \quad . \qquad [3.14]$$

The vector, r_l, calculated in [3.13] or [3.14] may be input as $r_{l,\tau}$ in equation [3.1] for a subsequent calculation run.

The option exists to rescale the original PR matrix, $r_{l,\tau}$, by dividing by rmax. This gives a better idea of the relation of each year's values to the mean. Naturally, it is only valid for the interval $t = p,\ldots,q$.

Alternative calculation method:

This optional procedure for calculating population numbers, $N_{i,t}$, and fishing mortalities, $F_{l,\tau}$, does not require input values for fishing mortality in the last age, $F_{\mu,\tau}$.

Numbers at age ($i = b,\ldots,m$) for the last year ($t = t_f$) are calculated as in equation [3.2].

Equation [3.3] is then used to calculate numbers for ages $i = b,\ldots,m-1$ in year $t = t_f-1$. Fishing mortalities (equation [3.7]) for the same age interval and year are also computed.

The fishing mortality for the last age group ($i = m$) in year $t = t_f-1$ is estimated by the weighted mean mortality (equation [3.9]) for a user-specified interval of ages, $i = k,\ldots,l$. This estimate is then used to calculate the population number for that year and age group by equation [3.2].

Numbers and mortalities for the remaining years ($t = t_f-2, t_f-3,\ldots,t_0$) are found by repeating the process described in the last two paragraphs.

GENERAL REMARKS

The cohort analysis program is split into two partially identical sections (COHORT.BAS and COHORTC.BAS) in order to keep enough memory free for the potentially large matrices that are generated. When using large data sets (i.e. matrices approaching 20 ages by 20 years), care must be taken to minimize the number of calculated items present in memory at one time. This may be done by entering the plotting and fitting program which erases all calculated matrices or by rerunning the whole program which requires re-entry of input data.

All input matrices or vectors must be similar in dimension to the catch matrix. For example, if the catch matrix is 10 age-groups by 15 years, then the input fishing mortalities for the last year must have 10 age-groups, and the input fishing mortalities for the last age-group must contain 15 years of data. The catch matrix may be reduced in dimension by specifying an interval of ages or years to be used for the calculations. This reduction of dimensions is applied to all input and calculated data in the program. NOTE: any reduction to the size of the catch matrix requires that the other input options to which the reduction applies be rerun. If this is not done, then there is danger that the program will index incorrect locations in the arrays. The most common potential problem arises with input of the independent abundance index. Often, these independent indices cover smaller time or age spans than the catch matrix. In this situation the input array should be padded with blanks (interpreted by BASIC as zeros) to fill it to the size of the original catch matrix. See the sample printed output in Appendix 11 for an example.

All calculations are performed in double (16 digit) precision. It must, however, be noted that the actual precision of calculated output is only single precision. This is due to a shortcoming in the algorithm that IBM PC BASIC uses to perform the exponential function. This problem has apparently been rectified in the latest DOS release (version 2.0).

EXAMPLE TUTORIAL RUN

Execution begins with display of the input menu as below. Data for cod in 4VsW (Gray, 1979) are again used as an example. For the actual data, see Appendix 11.

```
                INPUT MENU FOR COHORT ANALYSIS

There are three main menus in this program: INPUT, CALCULATION, and
OUTPUT; each of which may be called from the others. If the OUTPUT
menu is invoked from the other menus, then a selection can be
made from a list of items presently defined and available for output.

F1.  Catch at age matrix (BEGIN WITH THIS OPTION)
F2.  Weight at age matrix or vector
F3.  Fishing mortality, F, for last year
F4.  Fishing mortality, F, for last age
F5.  Natural mortality, M (default is 0.2)
F6.  General title (default is blank) and increment of run number
F7.  F1-F6 in sequence
F8.  Independent index
F9.  List files on data diskette in Drive B
CTRL F2. CALCULATION MENU (WARNING: At least F1 and F3 must be defined)
CTRL F3. OUTPUT MENU
```

Items F1 and F9 all return to the input menu. We will begin by running through the input menu items.

F1. Catch at age matrix

This option must be run before all others as it determines the interval of ages and years that will be used in the analysis. Entering "N" to "Specify interval...?" instructs the program to use all elements along that dimension. The catch matrix shown in the Data Entry program example run is input, but only years 1971 through 1978 are selected. This is to match the research vessel data input to F8. Ages 1 through 12 are selected to match the interval that Gray (1979) uses in his cohort analysis.

```
Enter filename (including extension) of catch matrix:  4vswcod.cat

Data was stored in double precision.

Data goes from year 1966 to 1978
Specify interval of years? (Y/N)  y
Enter first year of interval:  1971
Enter last year of interval:  1978

Data goes from age 1 to 16
Specify interval of ages? (Y/N)  y
Enter first age of interval:  1
Enter last age of interval:  12

Interval specification will be automatically applied to the other
inputs. If a new interval specification is desired, the program
must be RUN again.

Strike any key to continue.
```

F2. Weight at age matrix or vector

This input is optional. In this example, a vector of weights at age was used and it was expanded as described below.

```
Enter filename (including extension) of weight at age matrix or vector:
4vswcod.wt

If the data is a vector, it will be expanded to a matrix of identical
columns with the same dimensions as the catch matrix.

Data was stored in single precision.

Strike any key to continue.
```

F3. Fishing mortality, F, for last year

This option contains considerable flexibility in what and how data is entered or modified. This is to facilitate the tuning process for terminal fishing mortality. The user may do a set of calculations with a given vector of mortalities, run the plot and fit program, return here, change a few values or the whole vector, and rerun the calculations until a suitable vector is achieved. A few ways of using this option to tune a cohort analysis are discussed below.

One approach is to use a pre-determined partial recruitment vector, r_{1,τ_i}, and multiply by a global fully recruited fishing mortality, F_{τ_i} (see equation [3.1]), which is varied on successive calculation runs. This is demonstrated below.

```
Use partial recruitment vector to calculate a vector for F, last year? (Y/N)
y

Enter filename (including extension) of partial recruitment vector: 4vswcod.pr

Data was stored in single precision.

Enter a global F value to multiply with the P.R. vector:  .18

Present title is :
F, last year (PR)

The general title entered in F6 (if selected) will be added to this.
Strike any key to continue.
```

On successive returns to this option, the user simply enters "Y" to the first question and enters a new trial global F value. NOTE: If a different PR vector is desired, then the program must be rerun (item F6 in the output menu and "RUN" again). CAUTION: If an arithmetic or geometric mean PR vector is calculated (item F10 in the calculation menu) then the PR vector used here is replaced by these new values. This may also be a useful tuning technique to find a stable vector based on historical partial recruitment.

For more information on the title, see item F6. It is displayed here so that the user knows whether or not partial recruitment is being used in the terminal fishing mortality vector ("PR" will appear in the title). Also, the run number is displayed (when F6 has been executed) to serve as a reminder to change this if a new run of calculations is desired.

A vector of fishing mortalities for the last year may also be entered directly without using partial recruitment. For the example below, the vector entered as "4vswcod.fly" happened to be the vector resulting from the sample run above. Also, a title has been defined.

```
Use partial recruitment vector to calculate a vector for F, last year? (Y/N)
n
Load fishing mortalities for last year from Diskette or enter/modify
by Keyboard? (D/K)   d

Enter filename (including extension) of fishing mortality vector:   4vswcod.fly

Data was stored in single precision.

Present title is :
F, last year : 4VsW Cod (RUN # 1)

Strike any key to continue.
```

If "K" had been entered in response to the second question, then, assuming the vector had not been previously defined the following would appear.

```
Enter the vector of mortalities corresponding in dimension to the
age interval you have specified in the catch matrix. Press return
after each value entered.
```

The values were entered (for consistency, the same vector as before) and the following appears. If the vector had been defined previously, the program would jump to this step without requesting keyboard entry of mortalities. Changes may be made as shown.

```
Present mortalities are:

     1      2      3      4      5      6      7      8      9     10
0.0023 0.0041 0.0180 0.0720 0.1800 0.1800 0.1620 0.1620 0.1620 0.1620

    11     12
0.1620 0.1620

Change a few values, Re-enter new vector, or Exit? (C/R/E)   c
Enter position of value to be changed:   12
Enter new value for position  12 :   .15
Change more values? (Y/N)   n

Present mortalities are:

     1      2      3      4      5      6      7      8      9     10
0.0023 0.0041 0.0180 0.0720 0.1800 0.1800 0.1620 0.1620 0.1620 0.1620

    11     12
0.1620 0.1500
```

```
Change a few values, Re-enter new vector, or Exit? (C/R/E)   e

Present title is :
F, last year : 4VsW Cod (RUN # 1)

Strike any key to continue.
```

F4. Fishing mortality, F, for last age

This menu item follows exactly the same algorithm as F3, except that no partial recruitment is included.

```
Load fishing mortalities for last age from Diskette or enter/modify
by Keyboard? (D/K)   d

Enter filename (including extension) of fishing mortality vector:   4vswcod.fla

Data was stored in single precision.

Present title is :
F, last age

The general title entered in F6 (if selected) will be added to this.
Strike any key to continue.
```

F5. Natural mortality, M

This option allows changes to be made to the natural mortality value used in the analysis. The default value is 0.2.

```
Current value for natural mortality is:     .2
Change? (Y/N)   n
```

F6. General title and increment run number

Any matrix or vector used as input or calculated in the program has a three component title associated with it. This is displayed on any output created. The first component is a specific title assigned by the program for each menu item as appropriate (eg. "F, last year", "N, at beg. year", "Total B, fishable"). This is separated by a colon from a general title (see below), which is followed by a run number. The run number may be incremented on successive runs during the tuning process.

```
Current general title is:     (blank)
Change? (Y/N)   y
Enter general title (<=45 chars.):
---------------------------------------------
4VsW Cod
Current run number is:     1
Change? (Y/N)   n
```

F7. F1-F6 in sequence

This option was included to save time when beginning an analysis. Input menu items F1 through F6 are executed sequentially without returning to the menu.

F8. Independent index

This option inputs the independent index of stock abundance to which the cohort analysis is tuned. The index may be a vector (not age-structured) or a matrix (age-structured). It is used as

the independent variable for the plotting and fitting program. Naturally, it must correspond in rank and dimension to the array against which it will be plotted. For more details, see item F4 in the output menu, and the documentation for the plotting ad fitting program. The index chosen here was smoothed (3-year moving average) 4+ population numbers from summer research cruise data.

```
Enter filename (including extension) of abundance index:  4vswcod.srv

Data was stored in single precision.

Strike any key to continue.
```

F9. List files on data diskette in Drive B

This option is useful for quickly displaying the files residing on the data diskette without having to exit the program. Here is a sample from the diskette containing the 4VsW cod data.

```
4VSWCOD .WT  4VSWCOD .PR  4VSWCOD .FLA 4VSWCOD .RV4 4VSWCOD .CPU 4VSWCOD .TC4
4VSWCOD .PRS 4VSWCOD .CAT 4VSWCOD .FLY 4VSWCOD .RV  4VSWCOD .SRV 4VSWCOD .PRC

Strike any key to continue.
```

CTRL F2. CALCULATION MENU

When the calculation menu is invoked for the first time after data input is complete, the following will appear.

```
Use the F's for the last age group entered in the Input menu or
have them Calculated from weighted F's for each year? (I/C)  i

One moment please. I am doing your cohort analysis calculations.
                                        Signed, IBMPC
```

The selection of the "I" option indicates that a conventional cohort analysis is to be performed. If the "C" option had been chosen then the analysis is done according to the alternative calculation method described in the Calculation Algorithm section. In this case, the user will be prompted to enter an age interval over which weighted mean fishing mortalities will be calculated.

At least the catch matrix and fishing mortalities for the last year and age must be defined for the "I" option in order to proceed with the calculations. The "C" option does not require fishing mortalities for the last age. If these requirements are not met then the program will automatically return to the input menu.

When the calculations (as described in the menu below) have been performed, selections may be made from the menu. These calculations are redone if any input data (except the independent index and general title) are changed.

```
        CALCULATION MENU FOR COHORT ANALYSIS

Numbers at age (beginning of year and midyear) and age-specific
fishing mortalities are automatically calculated for any given
set of input data when this menu appears. If a weight at age matrix
is present, then a midyear biomass matrix will be calculated.
All the options indicated exit to the output menu automatically.
Any action taken by the output menu will be on the variable
calculated in that option. If the output menu is directly invoked
from here, then a selection can be made from a list of items
presently defined and available for output.

F1.  Numbers at age, beginning of year
F2.  Numbers at age, midyear
F3.  Biomass, midyear or fishable
F4.  Age-specific fishing mortality
F5.  Total numbers, beginning of year
F6.  Total numbers, midyear
F7.  Total biomass (use only if F3 is present)
F8.  Mean fishing mortalities
F9.  Overall fishing mortalities
F10. Partial recruitment menu
CTRL F1. INPUT MENU
CTRL F3. OUTPUT MENU
```

As stated in the menu, all the calculation options exit to an output menu specific for what has been calculated in that option. Discussion of output is, however, reserved until later.

F1. Numbers at age, beginning of year

The program is routed to the output menu, as numbers at age were calculated before the calculation menu appeared.

```
Estimates for population numbers at age at the beginning of the year
have been calculated.
Press any key to display output menu.
```

F2. Numbers at age, midyear

Similar to F1.

F3. Biomass, midyear or fishable

Midyear biomass is already calculated when this item is chosen. This may be replaced by a fishable biomass matrix.

```
Midyear or Fishable biomass? (M/F)  m
```

Before fishable biomass can be calculated, however, a partial recruitment vector must be defined. This may be entered in input menu item F3 or computed in calculation menu item F10.

The remainder of this option is as F1.

F4. Age-specific fishing mortality

Similar to F1.

F5. Total numbers, beginning of year

The user may select an interval of ages over which to sum the population numbers by entering "Y", or sum over all ages by entering "N". The age interval over which the summing was done is appended to the general title in any output.

```
Specify interval of ages to be summed for beginning of year
population numbers? (Y/N)  y

Enter first age to be summed:  4
Enter last age to be summed:  12
Press any key to display output menu.
```

F6. Total numbers, midyear

Similar to F5.

F7. Total biomass

This option will sum whichever biomass matrix has been defined in F3 (midyear or fishable). Age intervals may be selected for total midyear biomass, but all ages are summed for total fishable biomass.

F8. Mean fishing mortalities

Calculated means may be weighted by numbers or unweighted. As above, age intervals may be selected.

```
Weighted or unweighted mean fishing mortalities? (W/U)  w
Specify interval of ages to be averaged for fishing mortalities? (Y/N)  y

Enter first age to be averaged:  5
Enter last age to be averaged:  12
Press any key to display output menu.
```

F9. Overall fishing mortalities

Again, an age interval may be specified. By the nature of the calculations the value for the last year in the data series is not defined. It is set to zero.

```
Specify interval of ages to be included in calculation
of overall fishing mortality? (Y/N)  n

NOTE: The value for 1978 is not defined. It has been set to zero.

Press any key to display output menu.
```

F10. Partial recruitment menu

When this option is invoked, the program pauses for calculations:

```
My silicon synapses are firing (calculating raw PR matrix)...
```

and then displays a menu.

```
The following options are available for manipulation of partial recruitments:

F1. Calculate standardized arithmetic mean PR vector for a specified
    interval of years.
F2. Divide PR matrix by value used to standardize vector calculated in F1.
F3. Calculate standardized geometric mean PR vector for a specified
    interval of years.
F4. Divide PR matrix by value used to standardize vector calculated in F3.
F5. Output raw PR matrix.

NOTE: Only the most recent option chosen will remain in memory for output.
```

F1: An interval of years must be specified for averaging over ages. This interval is appended to the general title in any output. Generally, the last year of data is not included in the interval. Observe the cautionary note.

```
Enter first year to be averaged:   1975
Enter last year to be averaged:    1977

Standardized partial recruitment vector has been calculated.
This vector may be used in input menu item 3 to generate a new F for last year.
NOTE: The vector just calculated has replaced the previous input vector.

Press any key to display output menu.
```

F2: Again, an interval of years must be specified. Even though the whole matrix is output, it is only valid for the specified interval.

F3: Similar to F1.

F3: Similar to F2.

F5: Proceeds directly to output menu.

CTRL F3. OUTPUT MENU

There are two nearly identical forms of the output menu. The general form allows the user to output any data that has been defined during the run of the program. The specific form is invoked after proceeding through a calculation menu item. It is through the specific menu that the user may enter the plotting and fitting program. Both forms of the menu are displayed below. Total midyear population numbers were used as an example of a specific output.

```
          OUTPUT MENU FOR COHORT ANALYSIS (GENERAL)

F1. Print on screen
F2. Print on line printer (will print filename if F3 was last item chosen
    and output menu was invoked from a calculation menu item)
F3. Output on diskette
F5. Go to data ENTRY program
F6. END program
CTRL F1. INPUT MENU
CTRL F2. CALCULATION MENU
```

```
        OUTPUT MENU FOR COHORT ANALYSIS (Total N, midyear)

F1. Print on screen
F2. Print on line printer (will print filename if F3 was last item chosen
    and output menu was invoked from a calculation menu item)
F3. Output on diskette
F4. Go to PLOT & FIT program (N.B. Only valid when coming from
    a calculation menu item. The last item executed will
    be passed to PLOT & FIT, as well as the independent index.
    Arrays defined in the calculation menu are then erased.)
F5. Go to data ENTRY program
F6. END program
CTRL F1. INPUT MENU
CTRL F2. CALCULATION MENU
```

F1. Print on screen

General: The format of the screen output is exactly the same as in the Edit option (F6) of the data entry program, except that no editing can be done. If, for some reason, editing is desired, then the data may be output to diskette (F3) and changed with the data entry program.

The user simply enters the two- or three-character code for the data to be displayed and the number of decimal places desired.

For the example menu shown below, all possible selections have been defined and are thus shown. If a particular menu item has not been defined, then it will not be shown here. CAUTION: Because of memory limitations, it is not recommended that all menu items be defined as they are here.

```
The following is available for screen display. The numbering is almost
the same as in the input menu (I) and calculation menu (C). To select
an item for printing, enter the code preceding the item (eg. C6).
Matrices with <=8 years of data will be printed on one screen,
but matrices with >8 years will need at least two screens. To
print the next screen, press 'Pg Up'. To reprint the previous
screen, press 'Pg Dn'. Press 'X' to exit to output menu. Each item
is printed with a title, ages and years (if applicable).


Enter code for item to be output:  c1
Enter desired number of decimal places for N at beg. year (0-4):  0

I1.  Catch at age matrix                  C1.  Numbers at age, beginning of year
I2.  Weight at age matrix                 C2.  Numbers at age, midyear
I3.  Fishing mortality, last year         C3.  Biomass, midyear or fishable
I4.  Fishing mortality, last age          C4.  Age specific fishing mortalities
I5.  Natural mortality                    C5.  Total numbers, beginning of year
I6.  Partial recruitment (input)          C6.  Total numbers, midyear
I7.  Independent index                    C7.  Total biomass, midyear or fishable
                                          C8.  Mean fishing mortalities
                                          C9.  Overall fishing mortalities
                                          C10. Partial recruitment (calc'd)
```

Specific: The user simply enters the desired number of decimal places.

F2. Print on line printer (See Appendix 11 for output of example data)

General: This option is very similar to the screen output. The major difference is that all the input menu items or all the calculation menu items may be printed at one time ("IA" or "CA"). If one of these options is selected, then the user may choose:

```
Default number of decimal places or Specify? (D/S):  d
```

The default option will print the output with no interruptions, while the specifying option will request the number of decimal places desired before printing each item.

```
The following is available for hardcopy. The numbering is almost the
same as in the input menu (I) and calculation menu (C). To select
an item for printing, enter the code preceding the item (eg. C6).
Each item is output with a title, ages and years (if applicable).
When using IA or CA option, default number of decimal places is
zero for any population numbers or biomass options; three for all
other output.


Enter code for item to be output:  ca

I1.  Catch at age matrix                    C1.  Numbers at age, beginning of year
I2.  Weight at age matrix                   C2.  Numbers at age, midyear
I3.  Fishing mortality, last year           C3.  Biomass, midyear or fishable
I4.  Fishing mortality, last age            C4.  Age specific fishing mortalities
I5.  Natural mortality                      C5.  Total numbers, beginning of year
I6.  Partial recruitment (input)            C6.  Total numbers, midyear
I7.  Independent index                      C7.  Total biomass, midyear or fishable
                                            C8.  Mean fishing mortalities
                                            C9.  Overall fishing mortalities
                                            C10. Partial recruitment (calc'd)
IA.  Print all input items                  CA.  Print all calculation items
```

The format of the output is the same as the Hardcopy option (F9) in the data entry program.

Specific: The user simply enters the desired number of decimal places.

F3. Output on diskette

General: As in output to the screen and printer, the user enters the code for what is to be output to diskette. The output created here has the dimensions specified by the intervals of years and ages entered in input menu item F1 (input of catch matrix), not the dimensions of the data before these intervals (if any) were specified. Note that natural mortality (I5) may not be written to diskette.

The procedure for output is exactly the same as in item F7 of the data entry program.

```
The following is available for diskette output. The numbering is almost
the same as in the input menu (I) and calculation menu (C). To select
an item for output, enter the code preceding the item (eg. C6).
Each item is output with a title, precision, number of ages,
number of years, first age and first year.



Enter code for item to be output:  c10

I1.  Catch at age matrix                    C1.  Numbers at age, beginning of year
I2.  Weight at age matrix                   C2.  Numbers at age, midyear
I3.  Fishing mortality, last year           C3.  Biomass, midyear or fishable
I4.  Fishing mortality, last age            C4.  Age specific fishing mortalities
                                            C5.  Total numbers, beginning of year
I6.  Partial recruitment (input)            C6.  Total numbers, midyear
I7.  Independent index                      C7.  Total biomass, midyear or fishable
                                            C8.  Mean fishing mortalities
                                            C9.  Overall fishing mortalities
                                            C10. Partial recruitment (calc'd)
```

Specific: Same as item F7 of data entry program.

F4. Go to PLOT & FIT program

As can be seen from the menus, this option is only available in the specific output mode. If no independent abundance index is defined, the program returns to the input menu. Otherwise, control is passed to the program PREPLOT.BAS (Appendix 7). Here, the calculated index from cohort analysis is assigned to the dependent variable for the plotting and fitting program. To reserve memory for the plotting and fitting program, PREPLOT.BAS also erases all the calculated data arrays. The input arrays remain defined to allow subsequent return to COHORT.BAS without repeating execution of the whole input menu.

Gray's (1979) choice of independent index (smoothed 4+ research vessel population estimates) for tuning the cohort analysis will be used here. We will, however, use midyear rather than beginning of year population estimates as the index calculated from cohort analysis.

When PREPLOT.BAS has been executed, it chains to PLOT.BAS.

F5. Go to data ENTRY program

Memory is cleared and ENTRY.BAS is loaded and begins execution.

F6. END program

User is returned to BASIC command level. See CTRL F2 of data entry program.

5. PLOTTING AND LINE-FITTING PROGRAM

PROGRAM NAME

PLOT.BAS (listing in Appendix 8)

PURPOSE

Plots graph(s) and fits various regression models to data passed from the cohort analysis program.

CALCULATION ALGORITHM FOR REGRESSION ANALYSIS

Four regression models are available in the program: simple linear, linear forced through the origin, linear using geometric mean, and quadratic. In the algorithm, the matrix notation for regression analysis will be used as much as possible. For more information see eg. Draper and Smith (1981).

Recall that for regression purposes, there are two vector inputs from the cohort analysis program: the calculated index and the independent index. We will call the column vector of calculated values Y, and the matrix of independent values X. β is the vector of parameters to be estimated and ε is the vector of errors.

In general, the regression model may be put as

$$Y = X\beta + \varepsilon \quad . \qquad [5.1]$$

We will now examine the specific models used by the program.

Simple Linear Regression

In the simple line regression case, we specifically assign $X = \begin{bmatrix} 1 & X_{11} \\ 1 & X_{12} \\ . & . \\ . & . \\ . & . \\ 1 & X_{1n} \end{bmatrix}$ where the X_{1i} are the n individual values of the independent index, and $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$ where β_0 is the intercept parameter and β_1 is the slope parameter. The model equation may then be stated as

$$Y_i = \beta_0 + \beta_1 X_{1i} + \varepsilon_i \quad . \qquad [5.2]$$

The program finds estimates of β, denoted by $b = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$, using

$$b = (X'X)^{-1} X'Y \quad . \qquad [5.3]$$

(The prime symbol (') and the exponent of -1 are standard notation for the transpose and inverse of a matrix, respectively.) The standard errors for b_0 and b_1, which we shall call S_{b_0} and S_{b_1}, are also calculated. The mechanics of this are shown after the analysis of variance (ANOVA) calculations below, as values calculated in the ANOVA are used to compute the standard errors.

The program produces a summary table as follows.

		t- values	
Coefficient	Std. Error	Computed	Table
b_0	S_{b_0}	$\lvert b_0/S_{b_0} \rvert$	$t_{n-2,1-\alpha/2}$
b_1	S_{b_1}	$\lvert b_1/S_{b_1} \rvert$	$t_{n-2,1-\alpha/2}$

If the computed t-values exceed the tabled values (t with n-2 degrees of freedom, α=.05 or .01 by user's choice) then the hypothesis, H: β_i=0 (i=0 or 1), is rejected.

After the summary table, the program produces a standard analysis of variance table.

Source	df	SS	MS	Computed F	Table F
Mean	1	$SS(Mean)=(\sum_i Y_i)^2/n$			
b_1/b_0	1	$SS(b_1/b_0)$	$MS(b_1/b_0)$	$F=MS(b_1/b_0)/s^2$	$F_{1,\nu,1-\alpha}$
Residual	ν=n-2	SS(Residual)	$MS(Residual)=s^2$		
Total	n	Y'Y			

Sums of squares (SS) are calculated as follows:

$$SS(Regression) = SS(b_1/b_0) = b'X'Y - SS(Mean). \quad [5.4]$$

$$SS(Residual) = Y'Y - SS(Regression) - SS(Mean). \quad [5.5]$$

Mean squares (MS) are always found by dividing the sum of squares by the associated degrees of freedom (df).

It is from the s^2 value that the standard errors of the regression parameters are calculated. The estimated variance-covariance matrix for b, $\hat{V}(b)$, is first found:

$$\hat{V}(b) = \hat{V}\begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = (X'X)^{-1}s^2. \quad [5.6]$$

The standard errors, S_{b_0} and S_{b_1}, are found by taking the square root of the diagonal elements of $\hat{V}(b)$.

The multiple correlation index, R^2, is computed by

$$R^2 = SS(Regression)/[Y'Y - SS(Mean)]. \quad [5.7]$$

This measures the proportion of the total variation about the mean, $\bar{Y}$, explained by the regression.

The simple correlation between the variables X_1 and Y is found by using

$$r_{X_1Y} = [\sum_i (X_{1i}-\bar{X}_1)(Y_i-\bar{Y})]/[\sum_i (X_{1i}-\bar{X}_1)^2 \sum_i (Y_i-\bar{Y})^2]^{1/2}, \quad [5.8]$$

where $\bar{X}_1$ and $\bar{Y}$ are the means for X_1 and Y, respectively. For simple linear regression, $r^2_{X_1Y} = R^2$.

Linear Regression Through Origin

For this model, the β_0 parameter is excluded from equation [5.2] (i.e. it is assumed to be zero). Thus, the matrix of independent values reduces to $X' = [X_1\ X_2\ \ldots\ X_n]$, and β becomes simply β_1, estimated by a different $b_1 = \sum_i X_i Y_i / \sum_i X_i^2$.

The summary table is of the same format as the previous section except that the line corresponding to b_0 is not included. The values in the table are calculated in the same fashion as before, however, the tabled t-value takes on n-1, as opposed to n-2, degrees of freedom to account for the lack of the β_0 parameter.

The ANOVA table is similarly reduced by removing the line for the mean and adding one degree of freedom to the residuals. The sum of squares due to regression (equation [5.4]) then becomes simply $SS(regression) = b_1X'Y = (\sum_i X_i Y_i)^2/\sum_i X_i^2$.

The use of this method is cautioned even if the simplifying assumption of setting β_0 to zero seems justified. This is especially true if the range of values are far from the origin. Using the full model of equation [5.2] may be more appropriate for this case. Ostle and Mensing (1975) summarize by saying that the model should be chosen only after proper consideration has been given to all factors involved (this, of course, may be applied to model selection in general).

Geometric Mean Estimate of the Functional Regression of Y on X

This model is described in Ricker (1973), and uses the principle of minimizing the sum of squares of the products of both the vertical and horizontal distance of each point to the line. The output format and calculations are the same as the simple linear regression case, except for the estimate of $\beta = \begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}$.

These are calculated as follows (using summation rather than matrix notation):

$$\hat{\beta}_1 = b_1 = \pm[\Sigma(Y_i-\bar{Y})^2/\Sigma(X_{1i}-\bar{X}_1)^2]^{1/2} \qquad [5.9]$$

$$\hat{\beta}_0 = b_0 = \bar{Y}-b_1\bar{X}. \qquad [5.10]$$

The sign in [5.9] is determined from signum $\Sigma(X_{1i}-\bar{X}_1)(Y_i-\bar{Y})$.

It should be noted that because only the standard deviations of X and Y are incorporated in the regression parameter, β_1, (i.e. no cross-products are involved) there is no strict validity to the test statistics that are generated (Ricker, 1973). Also, Ricker states that the confidence limits are asymmetrical but that ordinary symmetrical limits are a reasonable approximation.

Quadratic Regression

In this model an X^2 term is added to equation [5.2]. We will denote this term by X_2 so that $X_{2i}=X_{1i}^2$. The model equation is thus

$$Y_i = \beta_0+\beta_1X_{1i}+\beta_2X_{2i}+\varepsilon_i. \qquad [5.11]$$

The matrix of independent variables is $X' = \begin{bmatrix}1 & 1 & \dots & 1\\ X_{11} & X_{12} & \dots & X_{1n}\\ X_{21} & X_{22} & \dots & X_{2n}\end{bmatrix}$ and we estimate $\beta' = [\beta_0\ \beta_1\ \beta_2]$ by again using equation [5.3]. The summary table is similar to the simple linear case, except that it contains an extra line for b_2 and the tabled t-values have n-3 degrees of freedom.

The analysis of variance table is as follows:

Source	df	SS	MS	Computed F	Table F
Mean	1	$SS(Mean)=(\Sigma Y_i)^2/n$			
b_1/b_0	1	$SS(b_1/b_0)$	$MS(b_1/b_0)$	$F=MS(b_1/b_0)/s^2$	$F_{1,\nu,1-\alpha}$
$b_2/(b_1,b_0)$	1	$SS(b_2/b_1,b_0)$	$MS(b_2/b_1,b_0)$	$F=MS(b_2/b_1,b_0)/s^2$	$F_{1,\nu,1-\alpha}$
$(b_1,b_2)/b_0$	2	$SS(b_1,b_2/b_0)$	$MS(b_1,b_2/b_0)$	$F=MS(b_1,b_2/b_0)/s^2$	$F_{2,\nu,1-\alpha}$
Residual	$\nu=n-3$	SS(Residual)	$MS(Residual)=s^2$		
Total	n	Y'Y			

The sums of squares due to regression and residuals are calculated as follows:

$$SS(Regression) = SS(b_1,b_2/b_0) = b'X'Y - SS(Mean) \qquad [5.12]$$

$$SS(Residual) = Y'Y - SS(Regression) - SS(Mean). \qquad [5.13]$$

The two remaining sums of squares are used to generate F-values for a sequential test. The value for $SS(b_1/b_0)$ is the same as that found in equation [5.4] for the simple linear model. $SS(b_2/b_1,b_0)$ is found by subtraction:

$$SS(b_2/b_1,b_0) = SS(b_1,b_2/b_0) - SS(b_1/b_0). \qquad [5.14]$$

Standard errors for the estimates of β, and the correlation index, R^2, are computed by the same method as for the simple linear case. Simple correlations (r_{X_1Y}, r_{X_2Y}, $r_{X_1X_2}$) are found by making the appropriate substitutions in equation [5.8].

A measure of collinearity is calculated for the quadratic model. Collinearity is said to exist if there are one or more linear relationships between the predictor variables (here, X_1 and X_2), for each observation (Willan and Watts, 1978). The result of this is ill conditioning in the X'X matrix

(i.e. $|X'X|$ is very small). Willan and Watts give a measure of this collinearity to be $|C|^{1/2}$ where C is the independent variable correlation matrix. They call this value the overall design efficiency factor. It represents the overall predictability of the data set or the factor by which the size of the joint confidence region could have been reduced if an efficient orthogonal design had been used. Thus, collinearity is high, when $|C|^{1/2}$ is low (i.e. approaching zero). The maximum possible value for the design efficiency factor is unity.

For the quadratic model, the program uses the equation,

$$|C|^{1/2} = (1-r^2_{X_1X_2})^{1/2}. \qquad [5.15]$$

GENERAL REMARKS

The program uses the independent abundance index as the independent variable (X) and the index calculated from cohort analysis as the dependent variable (Y). The two variables may be of age-structured (matrix) or non-age-structured (vector) form but must have the same rank and dimension. For matrix data, intervals of ages may be chosen for plotting and fitting. However, no distinction is made between ages, i.e. the program uses matrix data essentially as a long vector of catenated rows. The example tutorial run will use vector data (4+ midyear population estimates versus smoothed 4+ research vessel population estimates) but use of the program with matrix data will also be discussed.

The plotting and fitting program contains error-handling routines that return the user to the most recently used menu in the event of an irrecoverable error. The error number and the line number where it occured are displayed for several seconds before returning to the menu. The BASIC reference manual should be consulted for error descriptions. Depending on the nature of the error, the subsequent operation of the program may be somewhat impaired (for example, a graph may be displayed with axes only). This can usually be alleviated by returning to the "Plot again" option (F2) in the "Plot menu" if a graph has already been successfully produced. If an error occurs before a plot has been successfuly made, the most likely problem is that the ranks of the two input variables do not agree (eg. the independent index is a vector while the index from cohort analysis is a matrix). The user should then return to the cohort analysis program and correct the situation.

The residuals program contains a similar error-handling routine.

EXAMPLE TUTORIAL RUN

If the input data is non-age-structured then the program will immediately display the menu below. If the input data is age-structured then the program enters the subroutine for selection of ages to be plotted (F7 in the menu below), and then displays the menu.

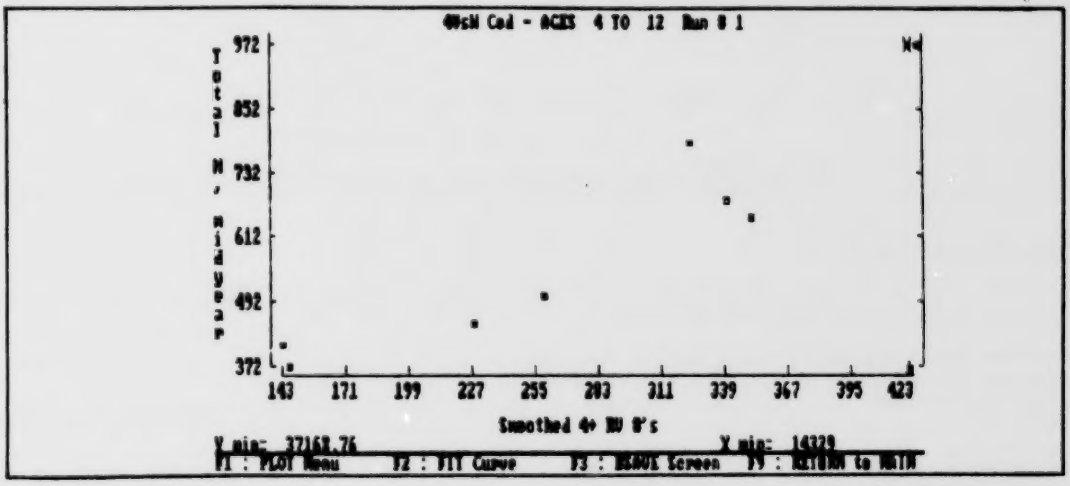
```
      >>>        PLOT   MENU        <<<

F1  : Automatic Scaling and PLOT (BEGIN WITH THIS OPTION)

F2  : PLOT again

F3  : Specify X Scale

F4  : Specify Y Scale

F5  : CHANGE Graph Title

F6  : CHANGE Axes Labels

F7  : Select AGES to be plotted

F9  : Return to MAIN PROGRAM

F10 : FIT a Curve

Press an F-key
```

The user begins with the automatic scaling and plot option (F1) and may then proceed with the other options in any order.

F1. Automatic Scaling and Plot

In this option, the program uses the minimum and maximum values of the data to scale the axes of the graph, and produces a preliminary plot on the screen. The various components of the graph (scaling and labelling of axes, graph title) will be discussed in the appropriate sections below (F3-F6).

Note that the minimum values for the X and Y data are displayed at the bottom of the graph. The numbers along both axes are scaled to give integer values less than 10,000. In this case, both X and Y axes have a scaling factor of 10^{-2}. The point marked by an X is for the last year of data.

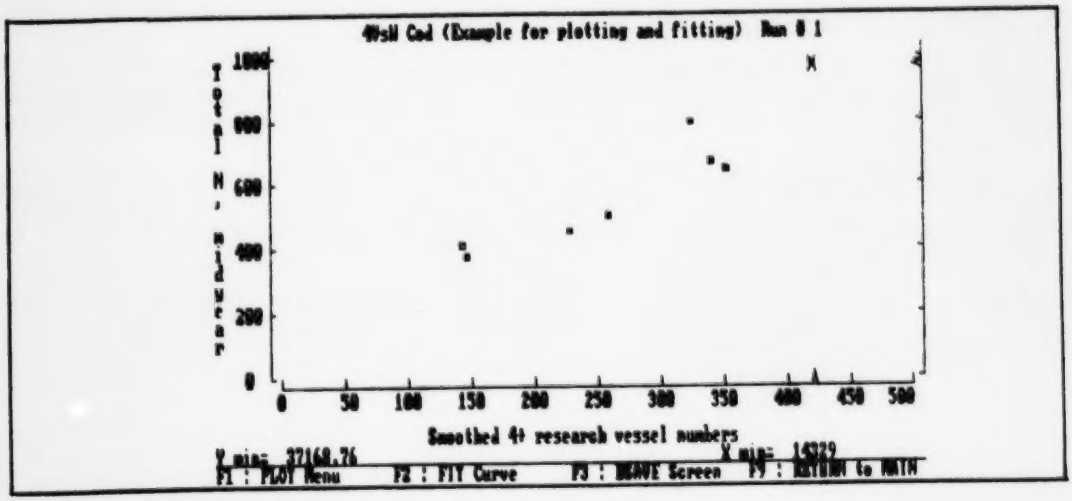


Note that several options are possible after the plot has been produced. The user may return to the plot menu (F1) for refinements to the graph, or may choose to fit a curve immediately (F2) by means of the fit menu. A copy of the screen buffer may be put to diskette by selecting the "BSAVE Screen" option (F3). The user is asked to enter a filename for the screen buffer file when this option is chosen. This file may be BLOAD'ed for later viewing (for details, see BASIC reference manual) or output to the printer with an appropriate software package (see Appendix 1). The "BSAVE Screen" option is available for all graphical output that the program produces. F9 returns the user to the input menu of the cohort analysis program.

F2. Plot again

This option is chosen to redisplay the graph after modifications have been made to it (eg. options F3-F6 described below). It can also be used after returning from the fit menu. Certain results from the calculations made in the automatic scaling option (F1) are required for this option, hence F1 must have been executed before (and every time the program is initiated from cohort analysis).

The displayed plot is what resulted after execution of menu items F3-F6.



F3. Specify X scale

The axes scaling options allow the user to define the value of the first tick-mark on the axis in question and the increment between ticks.

The first tick mark entered must be less than or equal to the minimum value displayed under "Current Values". The X-axis has ten tick-marks, hence the increment must be chosen so that ten times the increment plus the first tick value is greater than or equal to the maximum value. If these conditions are not met, the program asks for new values for the first tick and/or increment as appropriate. If the conditions are met, a new scaled axis is drawn with two triangles indicating the position of the minimum and maximum values. The user then presses the space bar if satisfied, or any other key if not (new first tick and increment may then be re-entered.) Again, the axes are scaled to give integer values less than 10,000. If scaling is done, this is indicated at the bottom of the screen, as it is in the example below. The carat (^) is the BASIC symbol for exponent.

The first and last ticks and tick increment in the "Current Values" section are scaled values.

```
     X - SCALE  SPECIFICATION

Current  Values :

min(x) = 14329                Max(x) = 42105

Scaling factor = 10 ^-2

First Tick = 143                          Last Tick = 423

Tick Increment = 28

>>> Enter NEW parameters, in SAME UNITS as data :

Enter 1st Tick ? 0

Enter new INCREMENT ? 5000
                                      NEW  X - AXIS :
                0  50 100 150 200 250 300 350 400 450 500
Is this correct ? (Space = OK)
*** LARGEST TICK HAS TOO FEW/MANY DIGITS ... TICK VALUES ARE SCALED DOWN :
*** NEW SCALING FACTOR = 10 ^-2
```

F4. Specify Y Scale

This option works in the same way as the scaling of the X-axis except that only five tick marks are used.

```
Y - AXIS :     >>> Y SCALE  SPECIFICATION  <<<

1000           Current  Values :
800            min(Y) = 37168.76      Max(Y) = 97069.66
600            Scaling factor = 10 ^-2
400            First Tick = 372    Last Tick = 972
200            Tick Increment = 120
0              >>> Enter NEW parameters, in SAME UNITS as data :
               Enter 1st Tick ? 0
               Enter new INCREMENT ? 20000


Is this correct ? (Space = OK)
*** LARGEST TICK HAS TOO FEW/MANY DIGITS ... TICK VALUES ARE SCALED DOWN :
*** NEW SCALING FACTOR = 10 ^-2
```

F5. Change graph title

The original title displayed by the plotting and fitting program is the general title output from the cohort analysis calculation menu item. This may include an interval of age groups used for totals or means (ages 4-12 in this example). The run number is also included if the length of the title is less than 70 characters.

Below is an example of how the user might change the title of the graph. A key besides the space bar was struck to indicate a title change.

```
>>>  CHANGE THE GRAPH TITLE  <<<

Current title =4VsW Cod - AGES  4 TO  12  Run # 1

NOTE :  Graph Title should be LESS THAN 70 chars long, if the 'RUN # ' is to be
appended.
CHANGE this title ?  (Space = NO Change)

Enter New Title :
!-------------------------------------------------------------------!
4VsW Cod (Example for plotting and fitting)
```

F6. Change axes labels

The axes labels may be changed in a fashion similar to the title. The original X axis label is the specific title associated with the independent index. The original Y axis label is the specific title associated with the cohort analysis calculation menu item.

Below is a demonstration of this option. The X axis label was changed (any key besides space bar) while the Y axis label was not (space bar). Observe the notes on label length.

```
>>> Change  the X-AXIS LABEL <<<

Current LABEL = Smoothed 4+ RV #'s

NOTE :  X-Label should be LESS THAN 70 chars long
CHANGE this title ?  (Space = NO Change)


Enter NEW LABEL :
:-------------------------------------------------------------:
Smoothed 4+ research vessel numbers




>>> Change  the Y-AXIS LABEL <<<

Current LABEL = Total N, midyear

NOTE :  Y-Label should be LESS THAN 20 chars long
CHANGE this title ?  (Space = NO Change)
```

F7. Select ages to be plotted

Selecting ages is only of use for matrix data (it makes no sense to select ages for non-age-structured indices!) As mentioned at the beginning of the tutorial, this option is automatically invoked before the plot menu if age-structured input is present. The example below is from an age-structured research vessel index.

```
The Input matrices have dimensions :   12  x 8

The FIRST AGE in the matrices is :  1

Ages currently selected :   FIRST AGE =  1
                             LAST AGE =  12

CHANGE these ages ? (Y/N)   Y

Select FIRST AGE ? 3

Select LAST AGE ? 8
```

F9. Return to main program

The user is returned to the input menu of the cohort analysis program where new runs of cohort analysis may be initiated. Recall that all data generated previously by the calculation menu was erased and must be rerun before entering the plot program again. The plotting and fitting program and its larger data arrays are erased from memory.

F10. Fit a curve

This option invokes the display of the line-fitting menu below. Option F9 is as immediately above, and F10 displays the plotting menu.

```
     >>>  FIT  MENU  <<<

F1  :  Linear Regression  (Simple Least Squares)

F2  :  Linear Regression  (Through Origin)

F3  :  Linear Regression  (Geometric Mean)

F4  :  Quadratic Regression

F7  :  Change significance level -- Current level = .05

F9  :  Return to MAIN Pgm.

F10 :  Return to PLOT Menu

Press an F-key
```

The four models outlined in the Calculation Algorithm are selected here. The first time a model is selected after coming from another menu, the following appears.

```
Input matrices range over years : 1971  to 1978

Do you wish to EXCLUDE A FEW YEARS from the Regression fitting ? Y/N
```

If the user selects "Y" then the last year to be used in the regression computations must be entered. The excluded years are, however, still plotted and used in the residual menu. The purpose of this exclusion of years is to minimize the effect of high sensitivity to input parameters in the tuning of the cohort analysis. For more background on the concept of sensitivity, see Rivard (1982) and Rivard and Doubleday (1979).

If the user selects "N" then all years will be used in the regression analysis. This was done for the examples below.

The same data will be used as an example for each of the four models. This is done solely for demonstration purposes as not all the models are appropriate. Again, it is the user's responsibility to choose a model suitable to his or her application.

F1. Linear regression (Simple least squares)

Below is a copy of the screen for the simple linear model. The "D" in the numbers is how BASIC signifies "ten to the exponent" for double precision values.

MODEL : Y = B0 + B1 X + e

Degree	Coefficient	Std. Error	t Values Computed	Table	(5%)
0	6.88383D+03	9.27806D+03	0.742	2.447	
1	1.92498D+00	3.17618D-01	6.061	2.447	

A N O V A

Source	df	SS	MS	Computed F	Table F
Mean	1	2.8941D+10			
b1 / b0	1	2.6001D+09	2.6001D+09	36.73	5.99
Residual	6	4.2472D+08	7.0787D+07		
Total	8	3.19659D+10			

R squared = 0.859589

Correlation Matrix : 1 0.92714
1

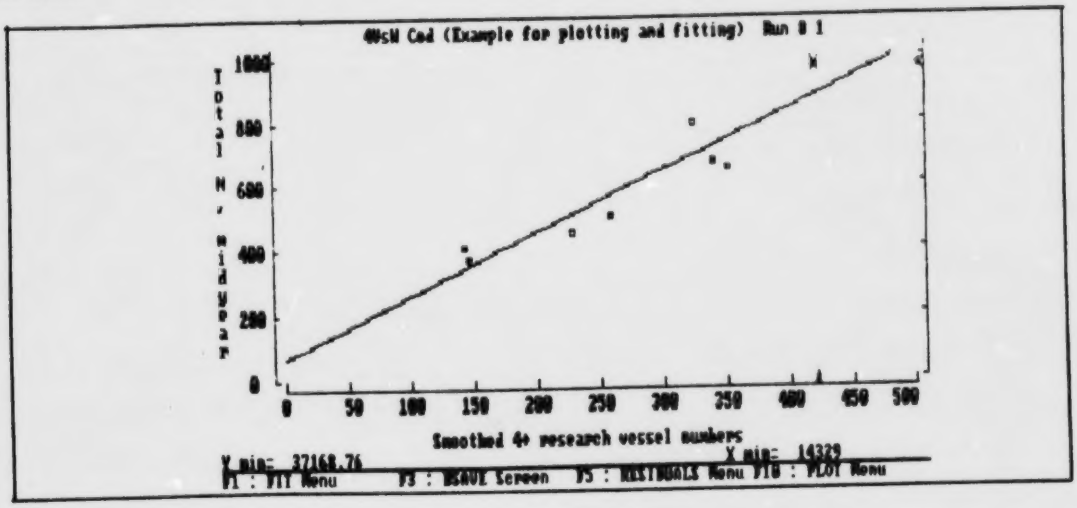
F1 PLOT Curve F10 > FIT Menu

To briefly summarize the output:

The estimated regression line is $\hat{Y}$ = 6883.33 + 1.982498X.
β_0 is not different from zero (computed t=0.742<table t=2.447) at 95% significance level.
β_1 (regression) is significantly different from zero (6.061>2.447 and computed F=36.73>table F=5.99).
R^2=0.859589 and r_{xy}=0.92714.
(The F and t table values are taken from one of four diskette files shown in Appendix 9.)

A copy of the screen is obtained by pressing the shift and PrtSc keys simultaneously.

Two options are available when the output has been displayed. The user may return to the fit menu (F10) or obtain a plot of the fitted line superimposed on the scatter diagram produced in the plot menu:

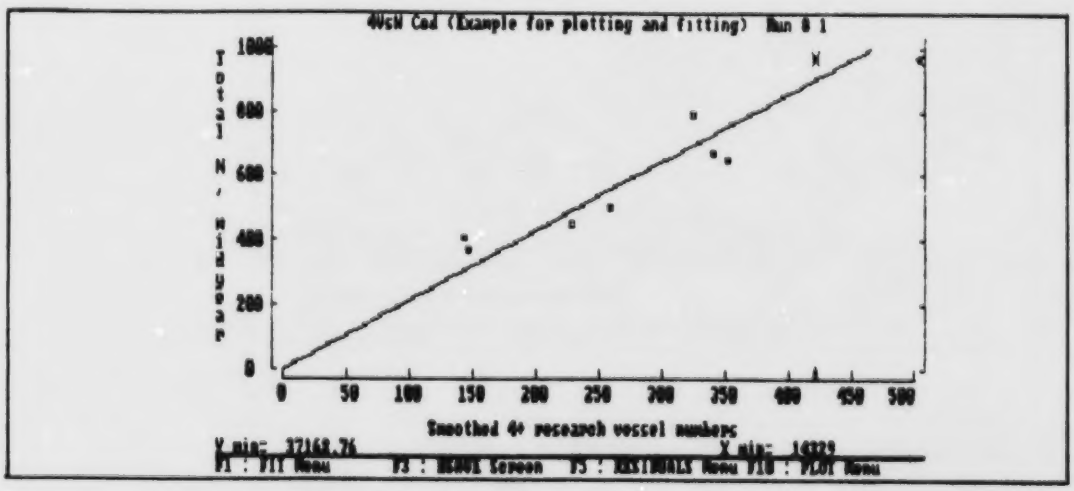

40cW Cod (Example for plotting and fitting) Run # 1
Total Nr. mid year
1000
800
600
400
200
0
0 50 100 150 200 250 300 350 400 450 500
Smoothed 4+ research vessel numbers
Y min= 37168.76
X min= 14329
F1 : FIT Menu F3 : RESCALE Screen F5 : RESIDUALS Menu F10 : PLOT Menu

Again, several choices exist. The only one not mentioned previously is the option of going to the residuals menu (discussed in the next section).

F2. Linear regression (Through origin)

Format is similar to F1.

```
                    MODEL  :  Y =  B X + e  -- (Line through the ORIGIN)

                                                    t  Values
Degree    Coefficient      Std. Error      Computed      Table      (5%)
  1       2.14819D+00      9.85072D-02       21.807      2.364


                                A N O V A
Source          df        SS              MS            Computed F        Table F
Regression       1    3.1502D+10      3.1502D+10          475.57            5.59
Residual         7    4.6369D+08      6.6242D+07
-------------------------------------------------------------------------------
Total            8    3.19659D+10

R squared =  0.985494

Correlation Matrix :   1      0.92714
                       1

F1  PLOT Curve                                              F10  > FIT Menu
```



F3. Linear regression (Geometric mean)

Format is similar to F1.

MODEL : Y = B0 + B1 X + e — B1 = GEOMETRIC MEAN Estimate

Degree	Coefficient	Std. Error	t Computed	t Values Table	(5%)
0	2.69814D+03	6.68344D+03	0.404	2.447	
1	2.07625D+00	2.28796D-01	9.075	2.447	

A N O V A

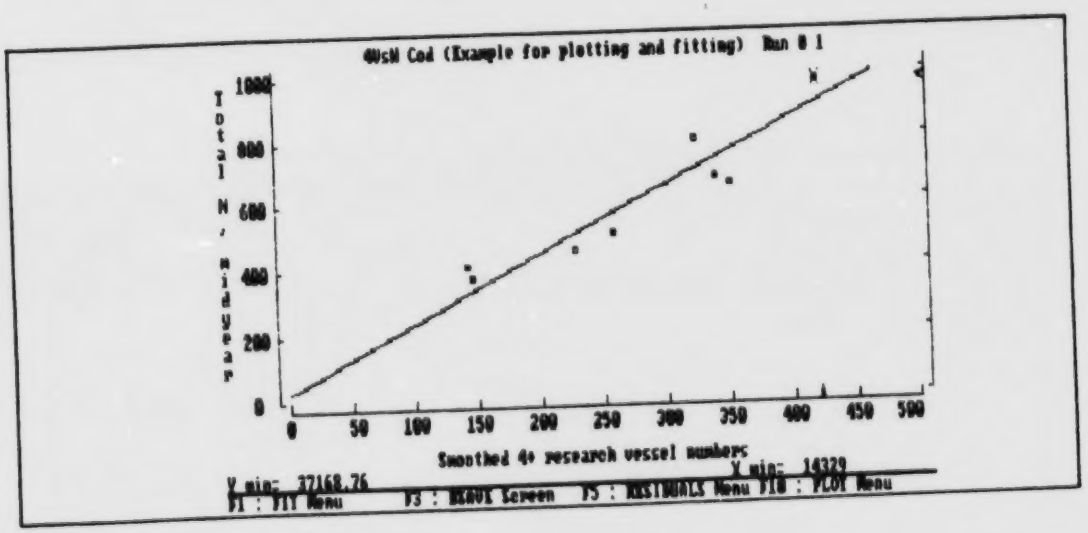
Source	df	SS	MS	Computed F	Table F
Mean	1	2.8941D+10			
b1 / b0	1	2.8045D+09	2.8045D+09	76.35	5.99
Residual	6	2.2039D+08	3.6732D+07		
Total	8	3.19659D+10			

R squared = 0.927140

Correlation Matrix : 1 0.92714
1

F10 > FIT Menu

F1 PLOT Curve



F4. Quadratic regression

Format is similar to F1. Note that the X^2 term is not significant to the model. This indicates that the simple linear model should be adequate (if all model assumptions such as $\varepsilon_i \sim$ Normal $(0,\sigma^2)$ are met.) The lack of significance of the X^2 term is also reflected by the low value (0.1497) for the design efficiency factor.

MODEL : Y = B0 + B1 X + B2 X^2 + e

Degree	Coefficient	Std. Error	t Values Computed	Table	(5%)
0	4.40039D+04	2.31110D+04	1.904	2.571	
1	-1.20046D+00	1.84431D+00	0.651	2.571	
2	5.78442D-05	3.37489D-05	1.714	2.571	

A N O V A

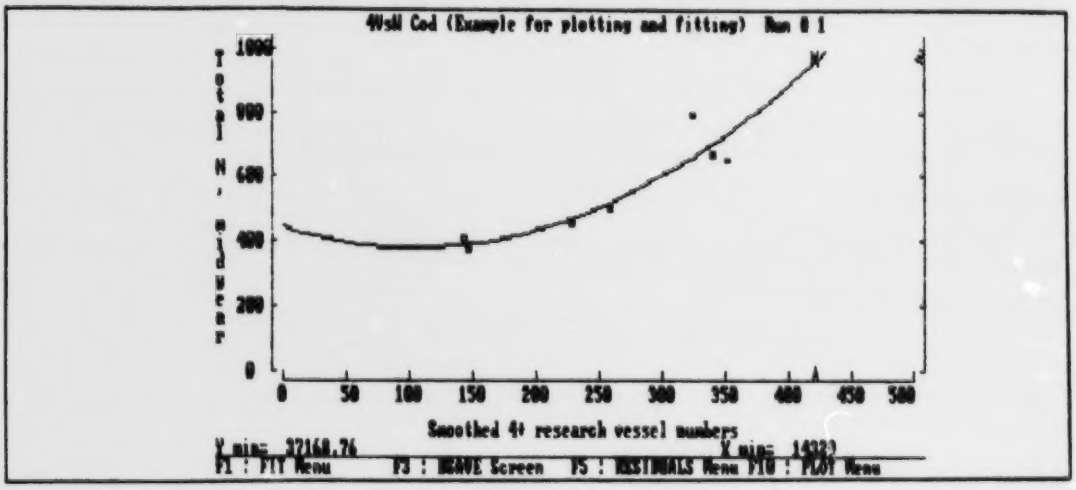
Source	df	SS	MS	Computed F	Table F
Mean	1	2.8941D+10			
b1 / b0	1	2.6001D+09	2.6001D+09	48.59	6.61
b2 /(b1,b0)	1	1.5718D+08	1.5718D+08	2.94	6.61
(b1,b2) /b0	2	2.7573D+09	1.3787D+09	25.77	5.79
Residual	5	2.6754D+08	5.3508D+07		
Total	8	3.19659D+10			

R squared = 0.911553

Correlation matrix :	Y	1	0.9271	0.9508
of variables	X		1	0.9887
	X^2			1

COLLINEARITY
Determinant = 0.0224
Sq. Root = 0.1497

F1 PLOT Curve F10 > FIT Menu



F7. Change significance level

The default significance level is α=0.05. This is displayed in the fit menu and when this option is selected. If a level of α=0.01 is desired, then enter "Y" as below. To change back to α=0.05, simply run through this menu item again.

```
CURRENT SIGNIFICANCE LEVEL = 5%

CHANGE THIS LEVEL ? (Y/N)
Y
```

6. RESIDUALS PROGRAM

PROGRAM NAME

RESIDS.BAS (listing in Appendix 10)

PURPOSE

Calculates, plots, and prints residuals for a given model selected in PLOT.BAS.

GENERAL REMARKS

The residuals are defined to be the n differences $e_i=Y_i-\hat{Y}_i$, $i=1,2,\ldots,n$ where Y_i is an observation and $\hat{Y}_i$ is the corresponding fitted value obtained by use of the fitted regression equation (Draper and Smith, 1981).

The program plots the residuals in four possible ways:

1. Against the X_j.
2. Against time (years).
3. Against observed Y values, Y_j.
4. Against predicted Y values, $\hat{Y}_j$.

The interpretation of these plots is beyond the scope of this report. For more information see eg. Draper and Smith (1981).

The residual plots are similar to the plots in PLOT.BAS, but there are a few differences. A reference line is drawn at $e_i=0$. No rescaling of axes may be done in this version of the program. The title is extended to include which residual plot has been selected. The extension of the title may cause the run number to be deleted (as it is here) or the last part of the title to be truncated.

A printout of the actual values is also available, both on the screen and on the printer.

As mentioned in the plot and fit documentation, this program contains an error handling routine to minimize the chance of having to repeat program execution unnecessarily.

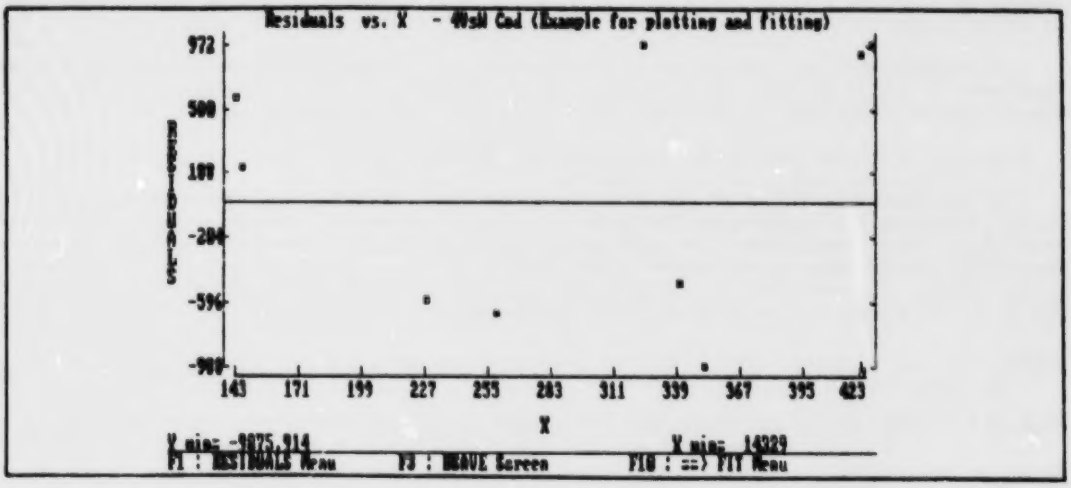
EXAMPLE TUTORIAL RUN

The residuals menu is invoked from the plot of a fitted line created in PLOT.BAS. For this example, the simple linear model is used.

```
>>>        RESIDUALS   MENU          <<<

F1  : PLOT against X

F2  : PLOT against YEARS

F3  : PLOT against OBSERVED Y

F4  : PLOT against PREDICTED Y

F7  : Display on SCREEN

F8  : PRINT Residuals

F10 : ==> FIT Menu

Press an F-key
```

F1. Plot against X

The only comment that will be made concerning the example plot is that few firm conclusions on the nature of the residuals can be made because of the small number of data points.



Plots of residuals versus years (F2), observed Y (F3), and predicted Y (F4) are displayed in a similar fashion.

F7. Display on screen, F8. Print residuals

Exactly the same thing is output on the screen and the printer. The only difference is for age-structured data where the residuals are displayed on the screen one age-group at a time, as opposed to the continuous output on the printer.

Below is a sample hardcopy.

RESIDUALS
4VsN Cod (Example for plotting and fitting) - RUN # 1

Year	I	Observed Y	Predicted Y	Residual
1971	3.2416E+04	7.8970E+04	6.9284E+04	9.6858E+03
1972	3.4024E+04	6.7484E+04	7.2379E+04	-4.8956E+03
1973	2.3903E+04	5.0028E+04	5.6746E+04	-6.7185E+03
1974	1.4329E+04	4.0928E+04	3.4467E+04	6.4611E+03
1975	1.4643E+04	3.7169E+04	3.5071E+04	2.0975E+03
1976	2.2793E+04	4.4871E+04	5.0760E+04	-5.8891E+03
1977	3.5142E+04	6.4635E+04	7.4531E+04	-9.8759E+03
1978	4.2105E+04	9.7070E+04	8.7935E+04	9.1347E+03

CONCLUSION

The report has focussed on the operation of a package of BASIC programs that allows tuning of a cohort analysis. It has not discussed specifically the various techniques used in calibrating cohort analysis results to independent indices of fish stock abundance (eg. adjusting terminal fishing mortalities to make the last year's point fall on the regression line, to make the regression line pass through the origin, or to maximize R^2 values.) It is the user's responsibility to define tuning criteria.

It is hoped that the flexibility of this package will not tempt the user to lose sight of the importance of a well-defined, objective approach to the tuning of cohort analysis. Rather, it should inspire investigation into wiser use of this important assessment tool.

ACKNOWLEDGEMENT

M. Joly was responsible for most of the programming in the plotting, line-fitting, and residuals portion of the package. The guidance of D. Rivard and Dr. W.G. Doubleday in the development and documentation of the package is much appreciated. Thanks also go to J. Longmuir for useful comments made in her review of the manuscript.

REFERENCES

Draper, N.R. and H. Smith. 1981. Applied Regression Analysis, 2nd ed. John Wiley & Sons, Inc., New York: 709 p.

Gray, D.F. 1979. 4VsW cod: background to the 1979 assessment. CAFSAC Res. Doc. 79/30: 35 p.

Ostle, B. and R.W. Mensing. 1975. Statistics in Research, 3rd ed. Iowa State Univ. Press, Ames: 596 p.

Pope, J.G. 1972. An investigation of the accuracy of virtual population analysis using cohort analysis. Int. Comm. Northwest Atl. Fish. Res. Bull. 9: 65-74.

Ricker, W.E. 1973. Linear regression in fishery research. J. Fish. Res. Board Can. 30: 409-434.

Rivard, D. 1982. APL programs for stock assessment (revised). Can. Tech. Rep. Fish. Aquat. Sci. 1091: 146 p.

Rivard, D. and W.G. Doubleday. 1979. Response of cohort analysis to input parameters. CAFSAC Res. Doc. 79/45: 26 p.

Willan, A.R. and D.G. Watts. 1978. Meaningful multicollinearity measures. Technometrics 20: 407-412.

APPENDIX 1. System Requirements

A. Hardware Configuration

The system used to develop the package was as follows:

IBM Personal Computer with 256K RAM
Two single-sided 160K 5¼" diskette drives with adapter
Amdek Color-II monitor
Color/graphics monitor adapter card
IBM 80 CPS Matrix Printer
Parallel printer adapter card.

There is flexibility in this configuration. The package should work with 128K (the BASIC segment should be 64K). The programs nearly fill one 160K diskette, so the diskette drives should not have a smaller capacity. However, larger capacity drives (with appropriate DOS) are acceptable. A color monitor is preferable but not essential (the programs have been run successfully with an NEC 12" green monochrome monitor). The monitor should, however, have graphics capability through a color/graphics monitor adapter card. Almost any dot matrix parallel printer compatible with the IBM and supporting compressed 132 character per line output is allowable (eg. Epson MX-series). An optional feature is a GRAFTRAX* ROM chip which allows graphics to be output to the printer.

B. Software Configuration

The programs were developed in advanced BASIC under Disk Operating System (DOS) Version 1.05. Previous (version 1.0) and newer (version 1.1) editions are also compatible.

An optional software package called PIXPRINT** may be used to output previously BSAVE'ed screen buffers of graphs to the printer.

* GRAFTRAX is a registered trademark of EPSON America, Inc.
** PIXPRINT is a registered trademark of MICROTAURE, Inc.

APPENDIX 2. Listing for AUTOEXEC.BAT (2 lines, 23 bytes).

```
date
basica SYSMENU
```

APPENDIX 3. Listing for SYSMENU.BAS (10 lines, 384 bytes).

```
10 CLS:PRINT ")))  COHORT  SYSTEM  ((("
20 PRINT :PRINT :PRINT "SELECT AN OPTION :"
30 PRINT :PRINT "1 - ENTRY  Pgm. :  To enter and store matrices as files."
40 PRINT :PRINT "2 - COHORT Pgm. :  To start the cohort computations"
50 PRINT :PRINT :PRINT "Enter a NUMBER "
60 K$ = INKEY$ : IF K$ = "" THEN 60
70 PRINT K$
80 IF K$ ="1" THEN CHAIN "ENTRY"
90 IF K$ ="2" THEN CHAIN "COHORT"
100 BEEP :GOTO 60
```

APPENDIX 4. Listing for ENTRY.BAS (631 lines, 19968 bytes without indentation*).

```
10 '*****DATA ENTRY PROGRAM*****PROGRAMMER: ROB KORVER******
20 OPTION BASE 1
30 TITLE1$=" " : TITLE2$=" "
40 FYEAR=1
50 FAGE=1
60 KEY OFF
70 SCREEN 0,0,0
80 ERASEFLAG=0
100 COLOR 2,0          'Menu
110 CLS
120 LOCATE  5,1,0
130 PRINT "              DATA ENTRY PROGRAM MENU"
140 PRINT
150 PRINT "Enter one choice by pressing the appropriately numbered F-key."
160 PRINT
170 PRINT "F1.  DIMENSIONS OF DATA (number of age groups, number of years)"
180 PRINT "F2.  TITLE (optional - default is blank)"
190 PRINT "F3.  FIRST YEAR OF DATA (optional - default is 1)"
200 PRINT "F4.  FIRST AGE GROUP (optional - default is 1)"
210 PRINT "F5.  DATA ENTRY"
220 PRINT "F6.  EDIT"
230 PRINT "F7.  DISKETTE OUTPUT"
240 PRINT "F8.  DISKETTE INPUT"
250 PRINT "F9.  HARDCOPY (NOTE: Filename will be printed if F7 or F8 immediately precede)"
260 PRINT "F10. EXPAND MATRIX (adds years to matrix presently in memory)"
270 PRINT "CTRL F1. START COHORT ANALYSIS"
275 PRINT "CTRL F2. END"
280   FOR I=1 TO 10 : KEY I, "" : NEXT I
290 C$=INKEY$ : IF C$="" THEN 290
300 CHECK1=ASC(C$)
310 LOCATE 1,1,0
320   IF CHECK1<>0 THEN PRINT CHR$(7) "Please press an F-key." : GOTO 290
330 EXTCODE=ASC(MID$(C$,2))
340   ON EXTCODE-58 GOSUB 1000,2000,3000,4000,5000,6000,8000,9000,7000,9500
350   IF EXTCODE<69 GOTO 100
360   IF EXTCODE=94 THEN CLEAR : CHAIN "COHORT"
365   IF EXTCODE=95 THEN GOTO 10000
370 PRINT CHR$(7) "Please press an F-key." : GOTO 290
1000 REM ***Enter dimensions***
1010 CLS : DISKFLAG=0
1020 PRINT "Entry of dimensions. Maximum 20 x 20 matrix."
1030 PRINT "NOTE: If data is a vector then enter '1' for the dimension"
1040 PRINT "not to be used."
1050 INPUT "Enter number of years:   ",NYEARS : IF NYEARS>20 THEN BEEP : GOTO 1050
1060 INPUT "Enter number of age groups:   ",NAGES : IF NAGES>20 THEN BEEP : GOTO 1060
1070 IF ERASEFLAG=1 THEN ERASE MATRIX$
1080 DIM MATRIX$ (NAGES,NYEARS)
1090 ERASEFLAG=1
1100 RETURN
2000 REM ***Enter title***
2010 CLS : DISKFLAG=0
2020 PRINT "Type of data is:    ";
2030 PRINT TITLE1$
2040 PRINT "Change? (Y/N)   ";
2050 CH$1$=INKEY$ : IF CH$1$="" THEN 2050
2060 PRINT CH$1$
2070   IF (CH$1$="Y") OR (CH$1$="y") THEN 2090
2080   IF (CH$1$="N") OR (CH$1$="n") THEN 2130 ELSE GOTO 2040
2090 PRINT "Enter type of data (<20 chars. - eg. Catch at age, Weight at age):   "
2100 PRINT STRING$(19,"-")
2110 LINE INPUT "",TITLE1$
2120   IF (LEN (TITLE1$))>19 THEN BEEP : GOTO 2090
```

* To conserve memory, the actual working copies of the programs do not use the indentation displayed here.

```
2130  PRINT
2140  PRINT "Identity of stock is:   " TITLE2$
2150  PRINT "Change? (Y/N)   ";
2160  CH$1A$=INKEY$ : IF CH$1A$="" THEN 2160
2170  PRINT CH$1A$
2180    IF (CH$1A$="Y") OR (CH$1A$="y") THEN 2200
2190    IF (CH$1A$="N") OR (CH$1A$="n") THEN RETURN ELSE GOTO 2150
2200  PRINT "Enter identity of stock (<=45 chars.):   "
2210  PRINT STRING$(45,"-")
2220  LINE INPUT "",TITLE2$
2230    IF (LEN (TITLE2$))>45 THEN BEEP : GOTO 2200
2240 RETURN
3000 REM ***Enter first year of data***
3010  CLS : DISKFLAG=0
3020  PRINT "Current first year is:   ";FYEAR
3030  PRINT "Change? (y/n)   ";
3040  CH$2$=INKEY$ : IF CH$2$="" THEN 3040
3050  PRINT CH$2$
3060    IF (CH$2$="Y") OR (CH$2$="y") THEN INPUT "Enter first year of data:   ",FYEAR : RETURN
3070    IF (CH$2$="N") OR (CH$2$="n") THEN RETURN ELSE GOTO 3030
4000 REM ***Enter first age***
4010  CLS : DISKFLAG=0
4020  PRINT "Current first age group is:   ";FAGE
4030  PRINT "Change? (y/n)   ";
4040  CH$3$=INKEY$ : IF CH$3$="" THEN 4040
4050  PRINT CH$3$
4060    IF (CH$3$="Y") OR (CH$3$="y") THEN INPUT "Enter first age group:   ",FAGE : RETURN
4070    IF (CH$3$="N") OR (CH$3$="n") THEN RETURN ELSE GOTO 4030
5000 REM ***Data entry***
5010  CLS : DISKFLAG=0
5020  PRINT "Data is entered by year, pressing return after each number entered."
5030  PRINT "The year and ages are printed across the top and down the left"
5040  PRINT "side of the screen, respectively. The user enters the data in"
5050  PRINT "the appropriate location. If a mistake is made in entering the"
5060  PRINT "data, use the cursor movement keys to move to the error and"
5070  PRINT "type over it. The backspace key and space bar may also be used."
5080  PRINT "Make sure to begin each entry in the first field on the left of"
5090  PRINT "the dark portion of the screen. To enter the next year of data,"
5100  PRINT "press the letter N at any time. Pressing N after the last year of"
5110  PRINT "data has been entered will return the user to the menu. If the"
5120  PRINT "user wishes to return to the menu without entering data for all"
5130  PRINT "years (eg. if very bad entry errors are made), then press N"
5140  PRINT "successively to advance the screen through the remaining years."
5150  PRINT "Only 9 digits including decimals and negative signs are allowed."
5160  PRINT
5170  PRINT "Press any key to begin data entry."
5180  B$=INKEY$ : IF B$="" THEN 5180
5200    FOR I=1 TO NYEARS
5210      INYEAR=FYEAR+I-1
5220      CLS
5230      COLOR 1,7
5240      PRINT SPACE$(79)
5250      LOCATE 1,7 : PRINT "YEAR" INYEAR
5260      IF NAGES=1 THEN PRINT "      " ELSE PRINT "  AGE "
5270      PRINT "      "
5280        FOR A=1 TO NAGES
5290          INAGE=FAGE+A-1
5300          IF NAGES=1 THEN PRINT "    "; ELSE PRINT USING "####":INAGE;
5310          PRINT "  "
5320        NEXT A
5330      COLOR 7,0
5340      GOSUB 5400
5350    NEXT I
5360 RETURN
```

```
5400 REM ***Enter one year of data***
5410  COLOR 2,0
5420  R=1
5430  C=1
5440  ENTRYMODE=-1
5450    WHILE ENTRYMODE
5460      ROW=R+3
5470      COL=C+7
5480      LOCATE ROW,COL,1
5490      E$=INKEY$ : IF E$="" THEN 5490
5500        IF (E$="N") OR (E$="n") THEN GOSUB 5600 : GOTO 5570
5510        IF E$=CHR$(13) THEN GOSUB 5620 : GOTO 5570   'E$=return
5520      ECODE=ASC(E$)
5530        IF (ECODE=43) OR (ECODE=45) OR (ECODE=46) OR (ECODE=32) OR (ECODE)47) AND (ECODE(58) THEN GOSUB 5660 : GOTO 5570
            'E$=valid character
5540        IF ECODE=8 THEN GOSUB 5690 : GOTO 5570       'E$=backspace
5550        IF ECODE=0 THEN GOSUB 5800 : GOTO 5570       'E$=extended code
5560      BEEP
5570    WEND
5580 GOSUB 5900        'Read screen into data matrix.
5590 RETURN
5600 REM ***Next year***
5610  ENTRYMODE=0 : RETURN
5620 REM ***Return***
5630  C=1
5640  IF R(NAGES THEN R=R+1 : RETURN
5650  BEEP : RETURN
5660 REM ***Print character on screen***
5670  IF C(=9 THEN PRINT E$ : C=C+1 : RETURN
5680  BEEP : RETURN
5690 REM ***Backspace***
5700  IF COL=8 THEN BEEP : RETURN
5710  PRINT " "
5720  C=C-1 : COL=C+7
5730  LOCATE ROW,COL : PRINT " "
5740 RETURN
5800 REM ***Extended code***
5805  EXCODE=ASC(MID$(E$,2))
5810    IF EXCODE=72 THEN GOSUB 5840 : GOTO 5830 'Cursor up
5815    IF EXCODE=80 THEN GOSUB 5855 : GOTO 5830 'Cursor down
5820    IF EXCODE=75 THEN GOSUB 5870 : GOTO 5830 'Cursor left
5825    IF EXCODE=77 THEN GOSUB 5885 : GOTO 5830 'Cursor right
5830 RETURN
5840 REM ***Cursor up***
5845  IF R)1 THEN R=R-1 : RETURN
5850  BEEP : RETURN
5855 REM ***Cursor down***
5860  IF R(NAGES THEN R=R+1 : RETURN
5865  BEEP : RETURN
5870 REM ***Cursor left***
5875  IF C)1 THEN C=C-1 : RETURN
5880  BEEP : RETURN
5885 REM ***Cursor right***
5890  IF C(9 THEN C=C+1 : RETURN
5895  BEEP : RETURN
5900 REM ***Read screen into data matrix***
5905    FOR AGE=1 TO NAGES
5910      READNUM$=""
5915      ROWM=AGE+3
5920      COLM=8
5925      R$=CHR$(SCREEN(ROWM,COLM))
5930        WHILE R$<>" "
5935          READNUM$=READNUM$+R$
5940          COLM=COLM+1
5945          R$=CHR$(SCREEN(ROWM,COLM))
5950        WEND
5955    MATRIX(AGE,I)=VAL(READNUM$) : NEXT AGE
5960 RETURN
```

```
6000 REM ***Edit subroutine***
6002  CLS : DISKFLAG=0
6004  PRINT "Up to 8 years of data can be displayed on the screen for editing"
6006  PRINT "at one time. To get the next 8 years press 'Pg Up'. To get the"
6008  PRINT "previous 8 years, press 'Pg Dn'. To move the cursor to the desired
6010  PRINT "position for corrections, use ";
6012  PRINT CHR$(24)+", "+CHR$(25)+", "+CHR$(26)+", "+CHR$(27);
6014  PRINT ". To move from a"
6016  PRINT "given field (year) to the beginning of the next or previous one,"
6018  PRINT "use 'Ctrl' + ";
6020  PRINT CHR$(26)+" or 'Ctrl' + "+CHR$(27);
6022  PRINT " respectively."
6024  PRINT "To change a number, merely type over it. The backspace and return"
6026  PRINT "keys, as well as the space bar may also be used. Only the matrix or"
6028  PRINT "vector portion of the data may be changed in this sub-routine."
6030  PRINT "For corrections to title, first age, and first year, the appro-"
6032  PRINT "priate sub-routine must be (re)invoked. To exit this routine,"
6034  PRINT "press the letter 'X' at any time."
6036  PRINT
6038  PRINT "Enter desired number of decimal places (0-4):   ";
6050  DECI$=INKEY$ : IF DECI$="" THEN 6050
6055  PRINT DECI$ : DECI=VAL(DECI$)
6060    IF DECI=0 THEN SOFTPR$="########" : GOTO 6070
6065    IF (DECI>0) AND (DECI<=4) THEN SOFTPR$=STRING$(8-DECI,"#")+"."+STRING$(DECI,"#") ELSE GOTO 6038
6070  PAGE=1 : OUTFLAG=0
6075  PAGES=CINT((HYEARS/8)+.45)
6080    IF PAGE=1 THEN GOSUB 6930 : GOSUB 6100 : GOTO 6095
6085    IF PAGE=2 THEN GOSUB 6950 : GOSUB 6100 : GOTO 6095
6090    IF PAGE=3 THEN GOSUB 6970 : GOSUB 6100 : GOTO 6095
6095    IF OUTFLAG=1 THEN RETURN ELSE GOTO 6080
6100 REM ***Print matrix and edit***
6105  LYR=LY+FYEAR-1
6110  HYR=HY+FYEAR-1
6115  COLOR 2,0
6120  CLS
6125  COLOR 1,7
6130  PRINT SPACE$(79);
6135  LOCATE 1,1,0 : PRINT TITLE1$; : IF (TITLE2$=" ") OR (TITLE2$="") THEN PRINT TAB(70) DATE$; ELSE PRINT " : "; TITLE2$ TAB(70)
      DATE$;
6140  COLOR 7,1
6145  LOCATE 2,1 : PRINT "  YEAR "
6150    FOR I=LY TO HY
6155      DUMMY=1
6160        IF LY=9 THEN DUMMY=DUMMY-8
6165        IF LY=17 THEN DUMMY=DUMMY-16
6170      PY=DUMMY+9-1
6175      PYR=FYEAR+I-1
6180      LOCATE 2,PY : PRINT USING "##########";PYR
6185    NEXT
6190  PRINT
6195    IF NAGES=1 THEN PRINT "       " ELSE PRINT "  AGE  "
6200    FOR J=1 TO NAGES
6205      IAGE=FAGE+J-1
6210        IF NAGES=1 THEN PRINT "    "; ELSE PRINT USING "####";IAGE;
6215        IF J=NAGES THEN PRINT "   "; ELSE PRINT "   "
6220    NEXT J
6225    FOR K=1 TO NAGES
6230      ROWE=K+4
6235      LOCATE ROWE,8
6240        FOR L=LY TO HY
6245          EVENODD=(INT(L/2))*2
6250            IF L=EVENODD THEN COLOR 0,2 ELSE COLOR 2,0
6255          PRINT USING SOFTPR$;MATRIX$(K,L);
6260    NEXT L,K
```

```
6300 REM ***Edit screen***
6305  FEELD=1 : LOCIF=1 : R=1 : EDITING=-1
6310    IF (PAGE<PAGES) AND (LY=9) THEN HY=8
6315    IF (PAGE=PAGES) AND (LY=9) THEN HY=NYEARS-8
6320    IF (PAGE=PAGES) AND (LY=17) THEN HY=NYEARS-16
6325    WHILE EDITING
6330      EVENODD=(INT(FEELD/2))*2
6335        IF FEELD=EVENODD THEN COLOR 0,2 ELSE COLOR 2,0
6340      ROWA=R+4
6345      COLA=LOCIF+(FEELD*9)-2
6350      LOCATE ROWA,COLA,1
6355      ED$=INKEY$ : IF ED$="" THEN 6355
6360        IF (ED$="I") OR (ED$="i") THEN GOSUB 6450 : GOTO 6395
6365        IF ED$=CHR$(13) THEN GOSUB 6460 : GOTO 6395     'ED$=return
6370      EDCODE=ASC(ED$)
6375        IF (EDCODE=43) OR (EDCODE=45) OR (EDCODE=46) OR (EDCODE=32) OR (EDCODE>47) AND (EDCODE<58) THEN NRFLAG=1 : GOSUB 6480 :
            GOTO 6395   'ED$=valid character
6380        IF EDCODE=8 THEN NRFLAG=1 : GOSUB 6505 : GOTO 6395   'ED$=backspace
6385        IF EDCODE=0 THEN GOSUB 6550 : GOTO 6395   'ED$=extended code
6390      BEEP
6395    WEND
6400    IF NRFLAG=1 THEN GOSUB 6800        'Read screen
6405  NRFLAG=0
6410 RETURN
6450 REM ***Finish editing***
6455  EDITING=0 : OUTFLAG=1 : RETURN
6460 REM ***Return***
6465  FEELD=1 : LOCIF=1
6470  IF R<NAGES THEN R=R+1 : RETURN
6475  BEEP : RETURN
6480 REM ***Print character on screen***
6485  IF FEELD=HY+1 THEN BEEP : RETURN
6490  IF (LOCIF=9) AND (FEELD<=HY) THEN PRINT ED$; : FEELD=FEELD+1 : LOCIF=1 : RETURN
6495  IF (LOCIF<9) AND (FEELD<=HY) THEN PRINT ED$; : LOCIF=LOCIF+1 : RETURN
6500  BEEP : RETURN
6505 REM ***Backspace***
6510  IF COLA=8 THEN PRINT " "; : BEEP : RETURN
6515  PRINT " "
6520  IF LOCIF=1 THEN FEELD=FEELD-1 : LOCIF=9 : RETURN
6525  IF LOCIF>1 THEN LOCIF=LOCIF-1
6530  RETURN
6550 REM ***Extended code***
6555  EXCODE=ASC(MID$(ED$,2))
6560    IF EXCODE=72  THEN GOSUB 6600 : RETURN      'Cursor up
6565    IF EXCODE=80  THEN GOSUB 6615 : RETURN      'Cursor down
6570    IF EXCODE=75  THEN GOSUB 6630 : RETURN      'Cursor left
6575    IF EXCODE=77  THEN GOSUB 6650 : RETURN      'Cursor right
6580    IF EXCODE=73  THEN GOSUB 6675 : RETURN      'Pg Up
6585    IF EXCODE=81  THEN GOSUB 6690 : RETURN      'Pg Dn
6590    IF EXCODE=115 THEN GOSUB 6705 : RETURN      'Previous field
6595    IF EXCODE=116 THEN GOSUB 6720 : RETURN      'Next field
6600 REM ***Cursor up***
6605  IF R>1 THEN R=R-1 : RETURN
6610  BEEP : RETURN
6615 REM ***Cursor down***
6620  IF R<NAGES THEN R=R+1 : RETURN
6625  BEEP : RETURN
6630 REM ***Cursor left***
6635  IF COLA=8 THEN BEEP : RETURN
6640  IF LOCIF=1 THEN FEELD=FEELD-1 : LOCIF=9 : RETURN
6645  LOCIF=LOCIF-1 : RETURN
6650 REM ***Cursor right***
6655  IF FEELD=HY+1 THEN BEEP : RETURN
6660  IF (LOCIF=9) AND (FEELD<=HY) THEN FEELD=FEELD+1 : LOCIF=1 : RETURN
6665  IF (LOCIF<9) AND (FEELD<=HY) THEN LOCIF=LOCIF+1 : RETURN
6670  BEEP : RETURN
```

```
6675 REM ***Page up***
6680  IF PAGE=PAGES THEN BEEP : RETURN
6685  PAGE=PAGE+1 : EDITING=0 : RETURN
6690 REM ***Page down***
6695  IF PAGE=1 THEN BEEP : RETURN
6700  PAGE=PAGE-1 : EDITING=0 : RETURN
6705 REM ***Previous field***
6710  IF FEELD>1 THEN FEELD=FEELD-1 : LOCIF=1 : RETURN
6715  BEEP : RETURN
6720 REM ***Next field***
6725  IF FEELD<NY THEN FEELD=FEELD+1 : LOCIF=1 : RETURN
6730  BEEP : RETURN
6800 REM ***Read screen***
6805    FOR AGE=1 TO NAGES
6810      ROWR=AGE+4
6815      YEARSUB=LY
6820        FOR FIELDR=1 TO NY
6825          READNUM$=""
6830            FOR LOCIFR=1 TO 9
6835              COLR=LOCIFR+(FIELDR*9)-2
6840              R$=CHR$(SCREEN(ROWR,COLR))
6845              READNUM$=READNUM$+R$
6850            NEXT LOCIFR
6855            IF (ASC(READNUM$))=37 THEN GOSUB 6900 : GOTO 6870
6860          MATRII$(AGE,YEARSUB)=VAL(READNUM$)
6865          YEARSUB=YEARSUB+1
6870    NEXT FIELDR,AGE
6875 RETURN
6900 FIELDR=NY+1 'Field overflow error
6905  LOCATE 1,1 : COLOR 4,7 : BEEP
6910  PRINT "Field overflow error: Please check the numbers for age ";
6915  PRINT AGE
6920    FOR I=1 TO 3500 : NEXT I
6925 RETURN
6930 REM ***Page 1***
6935  LY=1
6940  IF PAGE<PAGES THEN NY=8 ELSE NY=NYEARS
6945  RETURN
6950 REM ***Page 2***
6955  LY=9
6960  IF PAGE<PAGES THEN NY=16 ELSE NY=NYEARS
6965  RETURN
6970 REM ***Page 3***
6975  LY=17
6980  NY=NYEARS
6985  RETURN
7000 REM ***Hardcopy (HC) sub-routine***
7010  CLS
7020  LPRINT
7030  LPRINT
7040    IF DISKFLAG=1 THEN LPRINT CHR$(10) "Name of diskette file is:   "; FILESTORE$ : DISKFLAG=0 : FILESTORE$="" : LPRINT
7050  NAGESHC=NAGES
7060  NYEARSHC=NYEARS
7070  TITLE1HC$=TITLE1$ : TITLE2HC$=TITLE2$
7080  FYEARHC=FYEAR
7090  PAGEHC=PAGE
7100  DIM MATRIXHC$ (NAGESHC,NYEARSHC)
7110  DIMFLAG=1
7120    FOR I=1 TO NAGESHC
7130      FOR J=1 TO NYEARSHC
7140        MATRIXHC$(I,J)=MATRIX$(I,J)
7150    NEXT J,I
7200  PRINT "Enter desired number of decimal places (0-4):   ";
7210  DECIHC$=INKEY$ : IF DECIHC$="" THEN 7210
7220  PRINT DECIHC$ : DECIHC=VAL(DECIHC$)
7230    IF (DECIHC<0) OR (DECIHC>4) GOTO 7200
```

```
7240 MAXTESTHC$=-99999999999
7250   FOR I=1 TO NAGESHC
7260     FOR J=1 TO NYEARSHC
7270       IF ABS(MATRIXHC$(I,J)))>MAXTESTHC$ THEN MAXTESTHC$=ABS(MATRIXHC$(I,J))
7280   NEXT J,I
7290 MAXHC=LEN(STR$(INT(ABS(MAXTESTHC$))))-1
7300   IF DECIHC=0 THEN MAXLENHC=MAXHC+2 : GOTO 7320
7310 MAXLENHC=MAXHC+DECIHC+3
7320   IF MAXLENHC<5 THEN MAXLENHC=5
7330 NFIELDSHC=INT(126/MAXLENHC)
7340   IF DECIHC=0 THEN HARDPRHC$=STRING$(MAXLENHC,"#") : GOTO 7360
7350 HARDPRHC$=STRING$(MAXHC+2,"#")+"."+STRING$(DECIHC,"#")
7360 YEARPRHC$=STRING$(MAXLENHC,"#")
7370   IF NYEARSHC>NFIELDSHC THEN BLOCKSHC=2 ELSE BLOCKSHC=1
7380 WIDTH "LPT1:",132
7390   IF BLOCKSHC=1 THEN LYHC=1 : HYHC=NYEARSHC : GOSUB 7500 : WIDTH "LPT1:",80 : ERASE MATRIXHC$ : RETURN
7400   IF BLOCKSHC=2 THEN GOSUB 7800 : WIDTH "LPT1:",80 : ERASE MATRIXHC$ : RETURN
7500 REM ***Print data***
7510 LPRINT CHR$(15); TITLE1HC$; : IF (TITLE2HC$=" ") OR (TITLE2HC$="") THEN LPRINT TAB(120) DATE$ ELSE LPRINT " : " TITLE2HC$
     TAB(120) DATE$
7520 LPRINT
7530 LPRINT
7540 LPRINT CHR$(15); "YEAR ";
7550   FOR I=LYHC TO HYHC
7560     YEARHC=FYEARHC+I-1
7570     LPRINT USING YEARPRHC$;YEARHC;
7580   NEXT I
7590 LPRINT
7600 LPRINT
7610   IF NAGESHC=1 THEN LPRINT "     " ELSE LPRINT "AGE  "
7620 LPRINT
7630   FOR K=1 TO NAGESHC
7640     AGEHC=FAGEHC+K-1
7650       IF NAGESHC=1 THEN LPRINT "  "; ELSE LPRINT USING "##";AGEHC;
7660     LPRINT "   ";
7670       FOR L=LYHC TO HYHC
7680         LPRINT USING HARDPRHC$;MATRIXHC$(K,L);
7690       NEXT L
7700     LPRINT
7710   NEXT K
7720 LPRINT CHR$(18)
7730 RETURN
7800 REM ***Print data in two or more blocks***
7810 LYHC=1
7820 HYHC=NFIELDSHC
7830 GOSUB 7500
7840 LYHC=LYHC+NFIELDSHC
7850 HYHC=HYHC+NFIELDSHC
7860   WHILE HYHC<NYEARSHC
7870     GOSUB 7520
7880     LYHC=LYHC+NFIELDSHC
7890     HYHC=HYHC+NFIELDSHC
7900   WEND
7910 HYHC=NYEARSHC
7920 GOSUB 7520
7930 RETURN
8000 REM ***Disk output***
8010 CLS
8020 PRINT "Would you like single or double precision for diskette output? (S/D)   ";
8030 PRECISION$=INKEY$ : IF PRECISION$="" THEN 8030
8040 PRINT PRECISION$
8050   IF (PRECISION$="S") OR (PRECISION$="s") OR (PRECISION$="D") OR (PRECISION$="d") THEN 8060 ELSE GOTO 8020
8060   IF (PRECISION$="S") OR (PRECISION$="s") THEN GOSUB 8900
8070 GOSUB 8800
8080 INPUT "Enter filename:   ",FILENAME$
8090 PRINT "Insert data diskette into drive B."
```

```
8100  PRINT "Press any key to begin output to diskette."
8110  A$=INKEY$ : IF A$="" THEN 8110
8120  DOT=INSTR(FILENAME$,".")
8130    IF DOT=0 THEN FILESPEC$="B:"+FILENAME$+".DAT" ELSE FILESPEC$="B:"+FILENAME$
8140  ERRFLAG=1
8150 GOSUB 8400
8160    IF ERRFLAG=1 THEN GOSUB 8400
8170    OPEN FILESPEC$ FOR  OUTPUT AS #1
8180      PRINT#1, TITLE1$
8190      PRINT#1, TITLE2$
8200      PRINT#1, PRECISION$
8210      PRINT#1, NAGES;NYEARS;FASE;FYEAR;
8220        FOR I=1 TO NAGES
8230          FOR J=1 TO NYEARS
8240            IF (PRECISION$="D") OR (PRECISION$="d") THEN PRINT#1, MATRIX#(I,J); ELSE TEMP!=MATRIX#(I,J) : PRINT#1, TEMP!;
8250          NEXT J
8260        NEXT I
8270    CLOSE #1
8280    ON ERROR GOTO 0
8290  FILESTORE$=MID$(FILESPEC$,3)
8300  PRINT "Data has been stored on drive B as:   "FILESTORE$
8310  PRINT "Press any key to return to menu."
8320  DISKFLAG=1
8330  A$=INKEY$ : IF A$="" THEN 8330
8340 RETURN
8400 REM ***Check if file exists***
8410    ON ERROR GOTO 8500
8420  ERRFLAG=0
8430  LN=CSRLIN
8440  LOCATE 23,1 : COLOR 0,0
8450  FILECHECK=0
8460  FILES FILESPEC$ : FILECHECK=1
8470    IF FILECHECK=1 GOTO 8550    'File already exists
8480  COLOR 2,0 : LOCATE LN+1,1
8490    ON ERROR GOTO 8700 : RETURN
8500    IF ERR=53 THEN RESUME 8480
8510  COLOR 2,0 : LOCATE LN+1,1
8520  ON ERROR GOTO 0
8550 ON ERROR GOTO 0 : COLOR 2,0 : LOCATE LN+1,1
8555  ERRFLAG=1
8560  CLS
8565  PRINT CHR$(7) "File already exists. Would you like to: "
8570  PRINT "  A. Enter a different filename?"
8575  PRINT "  B. Rename the existing file without changing the name you have just entered?"
8580  PRINT "  C. Write over the existing file?"
8585  PRINT
8590  PRINT "Press A, B, or C on the keyboard.   ";
8595  OPT$=INKEY$ : IF OPT$="" THEN 8595
8600  PRINT OPT$
8605    IF (OPT$="A") OR (OPT$="a") THEN GOSUB 8620 : GOTO 8690
8610    IF (OPT$="B") OR (OPT$="b") THEN GOSUB 8640 : GOTO 8690
8615    IF (OPT$="C") OR (OPT$="c") THEN GOSUB 8680 : GOTO 8690 ELSE GOTO 8590
8620 INPUT "Enter new filename:   ",FILENAME$     '***Option A***
8625  DOT=INSTR(FILENAME$,".")
8630    IF DOT=0 THEN FILESPEC$="B:"+FILENAME$+".DAT" ELSE FILESPEC$="B:"+FILENAME$
8635 RETURN
8640 INPUT "Enter new filename for existing file:   ",NEWFN$  '***Option B***
8645  DOT=INSTR(NEWFN$,".")
8650    IF DOT=0 THEN NEWFS$="B:"+NEWFN$+".DAT" ELSE NEWFS$="B:"+NEWFN$
8655    ON ERROR GOTO 8750
8660  NAME FILESPEC$ AS NEWFS$
8665  NEWFS$=MID$(NEWFS$,3)
8670  PRINT "New name for existing file is:   "NEWFS$
8675 RETURN
8680 KILL FILESPEC$           '***Option C***
8685 RETURN
```

```
8690    ON ERROR GOTO 8700 : RETURN
8700 REM ***Errors for full diskette***
8710  CLS
8720    IF (ERR=61) THEN PRINT CHR$(7) "Data diskette is full." : PRINT "Insert a new diskette into drive B." : KILL FILESPEC$ :
        RESUME 8740
8730    IF (ERR=67) THEN PRINT CHR$(7) "Data diskette has too many files." : PRINT "Insert a new diskette into drive B." :
        RESUME 8740
8740    ON ERROR GOTO 0 : GOTO 8100
8750 REM ***NAME statement error***
8760    IF ERR=58 THEN PRINT CHR$(7) "New filename already exists." : RESUME 8640
8770    ON ERROR GOTO 0
8800 REM ***Instructions***
8810  PRINT
8820  PRINT "Rules for naming of files are as described in the BASIC reference"
8830  PRINT "manual. If no filename extension is included then the extension "
8840  PRINT "'.dat ' is added by default."
8850  PRINT
8860 RETURN
8900 REM ***Precision verification***
8910  PRINT CHR$(7)
8920  PRINT "WARNING: Single precision output will only store 7 significant digits."
8930  PRINT "Any numbers with more than 7 significant digits will be rounded off.     "
8940  PRINT "Press 'D ' if you would rather have double precision, if not, press RETURN."
8950  PRINT
8960  PV$=INKEY$ : IF PV$="" THEN 8960
8970    IF (PV$="D") OR (PV$="d") THEN PRECISION$="D" : RETURN
8980    IF (PV$=CHR$(13)) THEN RETURN ELSE GOTO 8940
9000 REM ***Diskette input***
9010  CLS
9020  PRINT "This sub-routine reads data already stored on diskette so that"
9030  PRINT "it may be verified and/or changed using the edit subroutine."
9040  PRINT "It may be rewritten to diskette with the output sub-routine."
9050  PRINT
9060  INPUT "Enter filename (including extension) of data:   ",FILESTORE$
9070  INFILESPEC$="B:"+FILESTORE$
9080  PRINT "Insert data diskette into drive B."
9090  PRINT "Strike any key when ready."
9100  A$=INKEY$ : IF A$="" GOTO 9100
9105    ON ERROR GOTO 9300
9110    OPEN INFILESPEC$ FOR INPUT AS #1
9120      LINE INPUT#1, TITLE1$
9130      LINE INPUT#1, TITLE2$
9140      LINE INPUT#1, PRECISION$
9150      INPUT#1, NAGES,NYEARS,FAGE,FYEAR
9160        IF ERASEFLAG=1 THEN ERASE MATRIX$
9170      ERASEFLAG=1
9180      DIM MATRIX$ (NAGES,NYEARS)
9190        FOR I=1 TO NAGES
9200          FOR J=1 TO NYEARS
9210            INPUT#1, MATRIX$(I,J)
9220        NEXT J,I
9230    CLOSE#1 : ON ERROR GOTO 0
9240  PRINT
9250    IF (PRECISION$="D") OR (PRECISION$="d") THEN PRINT "Data was stored in double precision." ELSE PRINT
        "Data was stored in single precision."
9260  PRINT "The file has been loaded. Press any key to return to menu."
9270  A$=INKEY$ : IF A$="" GOTO 9270
9280  DISKFLAG=1
9290 RETURN
9300 REM ***Trap 'File not found' error***
9310    IF ERR=53 THEN PRINT CHR$(7) : PRINT "File not found. Probable cause: wrong diskette and/or filename." :
        INPUT  "Re-enter filename:   ",FILESTORE$ : RESUME 9070
9320    ON ERROR GOTO 0
9500 REM ***Expand matrix***
9510  CLS
```

```
9520 PRINT "This option appends years previously stored on diskette to the matrix"
9530 PRINT "presently residing in memory."
9540 PRINT "NOTES: 1. Data to be added must have same number of age groups as present data."
9550 PRINT "       2. The expanded matrix must not have more than 20 years of data and"
9560 PRINT "          will replace the present matrix."
9570 PRINT "       3. The present title is not changed."
9580 PRINT : INPUT "Enter filename (including extension) of data to be appended:   ",FILESTORE$
9590 INFILESPEC$="B:"+FILESTORE$
9600 PRINT "Insert data diskette into drive B."
9610 PRINT "Strike any key when ready."
9620 A$=INKEY$ : IF A$="" GOTO 9620
9630   ON ERROR GOTO 9770
9640   OPEN INFILESPEC$ FOR INPUT AS #1
9650     LINE INPUT#1, TITLE1A$
9660     LINE INPUT#1, TITLE2A$
9670     LINE INPUT#1, PRECISION$
9680     INPUT#1, NAGESA,NYEARSA,FAGEA,FYEARA
9690     DIM MATRIXA$ (NAGESA,NYEARSA)
9700       FOR I=1 TO NAGESA
9710         FOR J=1 TO NYEARSA
9720           INPUT#1, MATRIXA$(I,J)
9730       NEXT J,I
9740   CLOSE#1 : ON ERROR GOTO 0
9750  PRINT
9760 GOTO 9800
9770 REM ***Trap 'File not found' error***
9780    IF ERR=53 THEN PRINT CHR$(7) : PRINT "File not found. Probable cause: wrong diskette and/or filename." :
        INPUT  "Re-enter filename:   ",FILESTORE$ : RESUME 9590
9790    ON ERROR GOTO 0
9800 REM ***Combine matrices***
9810    IF (NYEARS+NYEARSA)>20) THEN ERASE MATRIXA$ : PRINT CHR$(7) "There are more than 20 years in the combined matrix." :
        GOTO 9910
9820  DIM ADDMAT$(NAGES,NYEARS+NYEARSA)
9830    FOR I=1 TO NAGES : FOR J=1 TO NYEARS :ADDMAT$(I,J)=MATRIX$(I,J) : NEXT J,I
9840    FOR I=1 TO NAGES : FOR J=NYEARS+1 TO NYEARS+NYEARSA : ADDMAT$(I,J)=MATRIXA$(I,J-NYEARS) : NEXT J,I
9850  NYEARS=NYEARS+NYEARSA
9860  ERASE MATRIX$
9870  DIM MATRIX$(NAGES,NYEARS)
9880    FOR I=1 TO NAGES : FOR J=1 TO NYEARS :MATRIX$(I,J)=ADDMAT$(I,J) : NEXT J,I
9890  ERASE ADDMAT$,MATRIXA$
9900  PRINT "Expanded matrix has been created."
9910  PRINT "Press any key to return to menu."
9920  A$=INKEY$ : IF A$="" GOTO 9920
9930 RETURN
10000 REM ***Reset F-keys and end program***
10010  KEY 1,"list "
10020  KEY 2,"run"+CHR$(13)
10030  KEY 3,"load"+CHR$(34)
10040  KEY 4,"save"+CHR$(34)
10050  KEY 5,"cont"+CHR$(13)
10060  KEY 6,"edit "
10100  CLS
10200  KEY ON
10300 END
```

APPENDIX 5. Listing for COHORT.BAS (350 lines, 31104 bytes without indentation).

```
 5 '*****COHORT ANALYSIS PROGRAM (I/O)*****PROGRAMMER: ROB KORVER*****
 10  RUNNUM=1
 20  NMORT=.2 : OUTFLAGIS=1
 25  DIM DEFDECI(20),OUTVECI(20)
 30  KEY OFF
 40  DIM MATRIXOUT$(2,2)
 50  SCREEN 0,0,0
 60  DEFDECI(2)=3
 61  DEFDECI(3)=3
 62  DEFDECI(4)=3
 63  DEFDECI(6)=3
 64  DEFDECI(7)=3
 65  DEFDECI(11)=3
 66  DEFDECI(15)=3
 67  DEFDECI(16)=3
 68  DEFDECI(17)=3
 100 REM ***INPUT***
 105  COLOR 2,0
 110  CLS : DISKFLAG=0
 115  PRINT "                INPUT MENU FOR COHORT ANALYSIS"
 120  PRINT
 125  PRINT "There are three main menus in this program: INPUT, CALCULATION, and
 130  PRINT "OUTPUT; each of which may be called from the others. If the OUTPUT"
 140  PRINT "menu is invoked from the other menus, then a selection can be"
 150  PRINT "made from a list of items presently defined and available for output."
 160  PRINT
 165  PRINT "F1.  Catch at age matrix (BEGIN WITH THIS OPTION)"
 170  PRINT "F2.  Weight at age matrix or vector
 175  PRINT "F3.  Fishing mortality, F, for last year"
 180  PRINT "F4.  Fishing mortality, F, for last age"
 185  PRINT "F5.  Natural mortality, M (default is 0.2)"
 190  PRINT "F6.  General title (default is blank) and increment of run number"
 195  PRINT "F7.  F1-F6 in sequence
 210  PRINT "F8.  Independent index"
 215  PRINT "F9.  List files on data diskette in Drive B"
 220  PRINT "CTRL F2. CALCULATION MENU (WARNING: At least F1 and F3 must be defined)"
 230  PRINT "CTRL F3. OUTPUT MENU"
 250    FOR I=1 TO 10 : KEY I,"" : NEXT I
 255  C$=INKEY$ : IF C$="" THEN 255
 260  CHECK1=ASC(C$)
 265  LOCATE 25,1,0
 270    IF CHECK1<>0 THEN PRINT CHR$(7) "Please press a valid F-key."; : GOTO 255
 275  EXTCODE=ASC(MID$(C$,2))
 280    ON EXTCODE-58 GOSUB 1000,2000,3000,4000,5000,5200,6000,8000,9500
 290    IF EXTCODE-58<10 GOTO 100
 300    IF (EXTCODE=95) AND (CALCFLAG=1) THEN CHAIN "cohortc",10000,ALL
 310    IF (EXTCODE=95) AND (CALCFLAG=0) THEN CHAIN "cohortc",10600,ALL
 320    IF EXTCODE=96  GOTO 20200
 330  LOCATE 25,1,0 : PRINT CHR$(7) "Please press a valid F-key."; : GOTO 255
 1000 REM ***Input catch matrix***
 1010  CLS : CALCFLAG=1
 1020  INPUT "Enter filename (including extension) of catch matrix:   ",FILESTORE$
 1030 GOSUB 9000
 1040  I1TITLE1$=TITLE1$
 1050  I1TITLE2$=TITLE2$
 1055  PRINT : PRINT "Data goes from year" FYEAR "to" FYEAR+NYEARS-1
 1060  PRINT "Specify interval of years? (Y/N)   ";
 1070 QYRINT$=INKEY$ : IF QYRINT$="" THEN 1070
 1080  PRINT QYRINT$
 1090    IF (QYRINT$="Y") OR (QYRINT$="y") THEN INPUT "Enter first year of interval:   ", FYRINT : INPUT
       "Enter last year of interval:   ", LYRINT : GOTO 1110
 1100    IF (QYRINT$="N") OR (QYRINT$="n") THEN FYEAROUT=FYEAR : NYEARSOUT=NYEARS : LYEAR=0 :GOTO 1135 ELSE GOTO 1060
 1110  LYEAR=FYRINT-FYEAR
```

```
1120  NYEARSOUT=LYRINT-FYRINT+1
1130  FYEAROUT=FYRINT
1135  PRINT : PRINT "Data goes from age" FAGE "to" FAGE+NAGES-1
1140  PRINT "Specify interval of ages? (Y/N)   ";
1150  DAGEINT$=INKEY$ : IF DAGEINT$="" THEN 1150
1160  PRINT DAGEINT$
1170    IF (DAGEINT$="Y") OR (DAGEINT$="y") THEN INPUT "Enter first age of interval:   ", FAGEINT : INPUT
        "Enter last age of interval:   ", LAGEINT : GOTO 1190
1180    IF (DAGEINT$="N") OR (DAGEINT$="n") THEN FAGEOUT=FAGE : NAGESOUT=NAGES : LAGE=0 : GOTO 1220 ELSE GOTO 1140
1190  LAGE=FAGEINT-FAGE
1200  NAGESOUT=LAGEINT-FAGEINT+1
1210  FAGEOUT=FAGEINT
1220  IF EFLAGI1=1 THEN ERASE I1MATRI$
1230  EFLAGI1=1
1240  DIM I1MATRI$ (NAGESOUT,NYEARSOUT)
1250    FOR I=1 TO NAGESOUT
1260      FOR J=1 TO NYEARSOUT
1270        I1MATRI$(I,J)=MATRI$(LAGE+I,LYEAR+J)
1280    NEXT J,I
1285  PRINT
1290  PRINT "Interval specification will be automatically applied to the other"
1300  PRINT "inputs. If a new interval specification is desired, the program"
1310  PRINT "must be RUN again."
1320  PRINT
1330  PRINT "Strike any key to continue."
1340  A$=INKEY$ : IF A$="" THEN 1340
1350  OUTFLAGI1=1
1360  TITLE1OUT$=I1TITLE1$
1370  TITLE2OUT$=I1TITLE2$
1380  ERASE MATRIXOUT$
1390  DIM  MATRIXOUT$ (NAGESOUT, NYEARSOUT)
1400    FOR I=1 TO NAGESOUT
1410      FOR J=1 TO NYEARSOUT
1420        MATRIXOUT$(I,J)=I1MATRI$(I,J)
1430    NEXT J,I
1440 RETURN
2000 REM ***Input weight at age matrix***
2010  CLS : CALCFLAG=1
2020  INPUT "Enter filename (including extension) of weight at age matrix or vector:        ",FILESTORE$
2022  PRINT
2024  PRINT "If the data is a vector, it will be expanded to a matrix of identical"
2026  PRINT "columns with the same dimensions as the catch matrix."
2030 GOSUB 9000
2040  I2TITLE1$=TITLE1$
2050  I2TITLE2$=TITLE2$
2055  IF EFLAGI2=1 THEN ERASE I2MATRI$
2060  EFLAGI2=1
2061  DIM I2MATRI$ (NAGESOUT,NYEARSOUT)
2062    IF NYEARS=1 THEN 2063 ELSE GOTO 2070
2063    FOR I=1 TO NAGESOUT
2064      FOR J=1 TO NYEARSOUT
2065        I2MATRI$(I,J)=MATRI$(LAGE+I,1)
2066    NEXT J,I
2067 GOTO 2086
2070    FOR I=1 TO NAGESOUT
2075      FOR J=1 TO NYEARSOUT
2080        I2MATRI$(I,J)=MATRI$(LAGE+I,LYEAR+J)
2085    NEXT J,I
2086  PRINT
2087  PRINT "Strike any key to continue."
2088  A$=INKEY$ : IF A$="" THEN 2088
2090  OUTFLAGI2=1
2095  TITLE1OUT$=I2TITLE1$
2100  TITLE2OUT$=I2TITLE2$
2105  ERASE MATRIXOUT$
2110  DIM  MATRIXOUT$ (NAGESOUT, NYEARSOUT)
```

```
2115   FOR I=1 TO NAGESOUT
2120     FOR J=1 TO NYEARSOUT
2125       MATRIXOUT@(I,J)=I2MATRIX@(I,J)
2130   NEXT J,I
2135 RETURN
3000 REM ***Input fishing mortalities for last year***
3010  CLS : CALCFLAG=1
3020  J=1
3030 GOSUB 9700
3031  PRINT "Use partial recruitment vector to calculate a vector for F, last year? (Y/N)"
3033  PRF$=INKEY$ : IF PRF$="" THEN 3033
3034  PRINT PRF$
3035    IF (PRF$="Y") OR (PRF$="y") THEN 3037
3036    IF (PRF$="N") OR (PRF$="n") THEN 3049 ELSE 3031
3037    IF OUTFLAG1@=0 THEN GOSUB 7000 : VPY=1
3038  INPUT "Enter a global F value to multiply with the P.R. vector:   ",GLOBALF
3039  IF EFLAGI3=1 THEN ERASE I3MATRIX@
3040  EFLAGI3=1 : DIM I3MATRIX@(NAGESOUT,NYEARSOUT)
3041    FOR I=1 TO NAGESOUT : I3MATRIX@(I,1)=GLOBALF*IBMATRIX@(I,1) : NEXT
3042  I3TITLE1$="F, last year (PR)" : I3TITLE2$=GENTITLE$
3043 GOTO 3810
3049  PRINT "Load fishing mortalities for last year from Diskette or enter/modify"
3050  PRINT "by Keyboard? (D/K)   ";
3055  DK$=INKEY$ : IF DK$="" GOTO 3055
3060  PRINT DK$ : PRINT
3070    IF (DK$="D") OR (DK$="d") GOTO 3100
3080    IF (DK$="K") OR (DK$="k") GOTO 3300 ELSE GOTO 3049
3100  INPUT "Enter filename (including extension) of fishing mortality vector:   ", FILESTORE$
3110 GOSUB 9000 : J=1 : VPY=1
3120  I3TITLE1$=TITLE1$
3130  I3TITLE2$=GENTITLE$
3140  IF EFLAGI3=1 THEN ERASE I3MATRIX@
3150  EFLAGI3=1
3160  DIM I3MATRIX@ (NAGESOUT,NYEARSOUT)
3170    FOR I=1 TO NAGESOUT
3180      I3MATRIX@(I,J)=MATRIX@(LAGE+I,J)
3190    NEXT I
3200 GOTO 3800
3300 REM ***Keyboard entry***
3310  DISKFLAG=0
3320    IF VPY=1 GOTO 3500 ELSE GOTO 3330
3330  PRINT "Enter the vector of mortalities corresponding in dimension to the"
3340  PRINT "age interval you have specified in the catch matrix. Press return"
3350  PRINT "after each value entered."
3360  PRINT
3370  IF EFLAGI3=1 THEN ERASE I3MATRIX@
3380  EFLAGI3=1
3390  DIM I3MATRIX@ (NAGESOUT,NYEARSOUT)
3400    FOR I=1 TO NAGESOUT : INPUT " ",I3MATRIX@(I,J) : NEXT
3410  VPY=1
3500  PRINT
3510  PRINT "Present mortalities are:"
3520  PRINT
3530    IF NAGESOUT>10 GOTO 3570
3540    FOR I=1 TO NAGESOUT : PRINT USING "#######"; I; : NEXT
3545  PRINT
3550    FOR I=1 TO NAGESOUT : PRINT USING "##.####"; I3MATRIX@(I,J); : NEXT
3560  PRINT : GOTO 3650
3570    FOR I=1 TO 10 : PRINT USING "#######"; I; : NEXT
3580  PRINT
3590    FOR I=1 TO 10 : PRINT USING "##.####"; I3MATRIX@(I,J); : NEXT
3600  PRINT
3610  PRINT
3620    FOR I=11 TO NAGESOUT : PRINT USING "#######"; I; : NEXT
3625  PRINT
3630    FOR I=11 TO NAGESOUT : PRINT USING "##.####"; I3MATRIX@(I,J); : NEXT
```

```
3640  PRINT
3650  PRINT : PRINT "Change a few values, Re-enter new vector, or Exit? (C/R/E)   ";
3660  A$=INKEY$ : IF A$="" GOTO 3660
3665  PRINT A$
3670    IF (A$="C") OR (A$="c") GOTO 3700
3680    IF (A$="R") OR (A$="r") THEN VPY=0 : GOTO 3300
3690    IF (A$="E") OR (A$="e") GOTO 3800 ELSE GOTO 3650
3700  INPUT "Enter position of value to be changed:   ",POSCHG
3710  PRINT "Enter new value for position " POSCHG ":   ";
3720  INPUT "",I3MATRIX$(POSCHG,J)
3730  PRINT "Change more values? (Y/N)   ";
3740  A$=INKEY$ : IF A$="" GOTO 3740
3750  PRINT A$
3760    IF (A$="Y") OR (A$="y") GOTO 3700
3770    IF (A$="N") OR (A$="n") GOTO 3300 ELSE GOTO 3730
3800  I3TITLE1$="F, last year" : I3TITLE2$=GENTITLE$
3810  PRINT
3820  PRINT "Present title is :"
3830  PRINT I3TITLE1$;
3840    IF (I3TITLE2$=" ") OR (I3TITLE2$="") GOTO 3870 ELSE PRINT " : ";I3TITLE2$;" (RUN #";STR$(RUNNUM);")"
3842    IF OUTFLAG1$=1 THEN PRINT : GOTO 3890
3850  PRINT : PRINT "The general title entered in F$ (if selected) will replace
3860  PRINT "the portion after the colon." : GOTO 3890
3870  PRINT : PRINT
3880  PRINT "The general title entered in F$ (if selected) will be added to this."
3890  PRINT "Strike any key to continue."
3900  A$=INKEY$ : IF A$="" GOTO 3900
3910 GOSUB 9730
3920  OUTFLAG3=1
3930  TITLE1OUT$=I3TITLE1$
3940  TITLE2OUT$=I3TITLE2$
3950  ERASE MATRIXOUT$
3960  DIM MATRIXOUT$(NAGESOUT,1)
3970    FOR I=1 TO NAGESOUT : MATRIXOUT$(I,1)=I3MATRIX$(I,1) : NEXT
3980 RETURN
4000 REM ***Input fishing mortalities for last age***
4010  CLS : CALCFLAG=1
4020  I=1
4030 GOSUB 9750
4040  PRINT "Load fishing mortalities for last age from Diskette or enter/modify"
4045  PRINT "by Keyboard? (D/K)   ";
4050  DK$=INKEY$ : IF DK$="" GOTO 4050
4060  PRINT DK$ : PRINT
4070    IF (DK$="D") OR (DK$="d") GOTO 4090
4080    IF (DK$="K") OR (DK$="k") GOTO 4200 ELSE GOTO 4040
4090  INPUT "Enter filename (including extension) of fishing mortality vector:   ",FILESTORE$
4100 GOSUB 9000 : I=1 : VPA=1
4110  I4TITLE1$=TITLE1$
4120  I4TITLE2$=GENTITLE$
4130  IF EFLAGI4=1 THEN ERASE I4MATRIX$
4140  EFLAGI4=1
4150  DIM I4MATRIX$ (NAGESOUT,NYEARSOUT)
4160    FOR J=1 TO NYEARSOUT
4170      I4MATRIX$(I,J)=MATRIX$(I,LYEAR+J)
4180    NEXT J
4190 GOTO 4620
4200 REM ***Keyboard entry***
4210  DISKFLAG=0
4220    IF VPA=1 GOTO 4320 ELSE GOTO 4230
4230  PRINT "Enter the vector of mortalities corresponding in dimension to the"
4240  PRINT "year interval you have specified in the catch matrix. Press return
4250  PRINT "after each value entered."
4260  PRINT
4270  IF EFLAGI4=1 THEN ERASE I4MATRIX$
4280  EFLAGI4=1
4290  DIM I4MATRIX$ (NAGESOUT,NYEARSOUT)
```

```
4300   FOR J=1 TO NYEARSOUT : INPUT " ",I4MATRII0(I,J) : NEXT
4310 VPA=1
4320 PRINT
4330 PRINT "Present mortalities are:"
4340 PRINT
4350   IF NYEARSOUT>10 GOTO 4400
4360   FOR J=1 TO NYEARSOUT : PRINT USING "########"; J; : NEXT
4370 PRINT
4380   FOR J=1 TO NYEARSOUT : PRINT USING "##.####"; I4MATRII0(I,J); : NEXT
4390 PRINT : GOTO 4480
4400   FOR J=1 TO 10 : PRINT USING "########"; J; : NEXT
4410 PRINT
4420   FOR J=1 TO 10 : PRINT USING "##.####"; I4MATRII0(I,J); : NEXT
4430 PRINT
4440 PRINT
4450   FOR J=11 TO NYEARSOUT : PRINT USING "########"; J; : NEXT
4455 PRINT
4460   FOR J=11 TO NYEARSOUT : PRINT USING "##.####"; I4MATRII0(I,J); : NEXT
4470 PRINT
4480 PRINT : PRINT "Change a few values, Re-enter new vector, or Exit? (C/R/E)   ";
4490 A$=INKEY$ : IF A$="" GOTO 4490
4500 PRINT A$
4510   IF (A$="C") OR (A$="c") GOTO 4540
4520   IF (A$="R") OR (A$="r") THEN VPA=0 : GOTO 4200
4530   IF (A$="E") OR (A$="e") GOTO 4620 ELSE GOTO 4480
4540 INPUT "Enter position of value to be changed:   ",POSCH6
4550 PRINT "Enter new value for position " POSCH6 ":   ";
4560 INPUT "",I4MATRII0(I,POSCH6)
4570 PRINT "Change more values? (Y/N)   ";
4580 A$=INKEY$ : IF A$="" GOTO 4580
4590 PRINT A$
4600   IF (A$="Y") OR (A$="y") GOTO 4540
4610   IF (A$="N") OR (A$="n") GOTO 4320 ELSE GOTO 4570
4620 I4TITLE1$="F, last age" : I4TITLE2$=GENTITLE$
4630 PRINT
4640 PRINT "Present title is :"
4650 PRINT I4TITLE1$;
4660   IF (I4TITLE2$=" ") OR (I4TITLE2$="") GOTO 4690 ELSE PRINT " : ";I4TITLE2$;" (RUN #";STR$(RUNNUM);")"
4662   IF OUTFLAGI$=1 THEN PRINT : GOTO 4710
4670 PRINT : PRINT "The general title entered in F$ (if selected) will replace
4680 PRINT "the portion after the colon." : GOTO 4710
4690 PRINT : PRINT
4700 PRINT "The general title entered in F$ (if selected) will be added to this."
4710 PRINT "Strike any key to continue."
4720 A$=INKEY$ : IF A$="" GOTO 4720
4730 GOSUB 9780
4740 OUTFLAGI4=1
4750 TITLE1OUT$=I4TITLE1$ .
4760 TITLE2OUT$=I4TITLE2$
4770 ERASE MATRIIOUT$
4780 DIM MATRIIOUT$(1,NYEARSOUT)
4790   FOR J=1 TO NYEARSOUT : MATRIIOUT$(1,J)=I4MATRII0(1,J) : NEXT
4800 RETURN
5000 REM ***Change natural mortality***
5010 CLS : CALCFLAG=1
5020 PRINT "Current value for natural mortality is:   " MMORT
5030 PRINT "Change? (Y/N)   ";
5040 A$=INKEY$ : IF A$="" GOTO 5040
5050 PRINT A$
5060   IF (A$="Y") OR (A$="y") THEN INPUT "Enter new value for natural mortality:   ",MMORT : RETURN
5070   IF (A$="N") OR (A$="n") THEN RETURN ELSE GOTO 5030
5200 REM ***Input general title***
5210 CLS : OUTFLAGI$=1
5212 CHOPGTIT=INSTR(GENTITLE$,"(RUN #")
5214   IF CHOPGTIT<>0 THEN GENTITLE$=LEFT$(GENTITLE$,(CHOPGTIT-1))
5220 PRINT "Current general title is:   " GENTITLE$
```

```
5230  PRINT "Change? (Y/N)   ";
5240  A$=INKEY$ : IF A$="" GOTO 5240
5250  PRINT A$
5260    IF (A$="Y") OR (A$="y") GOTO 5270 ELSE GOTO 5310
5270  PRINT "Enter general title (<=45 chars.):   "
5280  PRINT STRING$(45,"-")
5290  LINE INPUT "",GENTITLE$
5300    IF (LEN(GENTITLE$))>45 THEN BEEP : GOTO 5270
5302  I3TITLE2$=GENTITLE$ : I4TITLE2$=GENTITLE$
5305 GOTO 5320
5310    IF (A$="N") OR (A$="n") THEN 5320 ELSE GOTO 5230
5320  PRINT "Current run number is:   " RUNNUM
5330  PRINT "Change? (Y/N)   ";
5340  A$=INKEY$ : IF A$="" GOTO 5340
5350  PRINT A$
5360    IF (A$="Y") OR (A$="y") THEN INPUT "Enter new run number:   ",RUNNUM : RETURN
5370    IF (A$="N") OR (A$="n") THEN RETURN ELSE GOTO 5330
6000 REM ***F1-F6 in sequence***
6100  GOSUB 1000
6200  GOSUB 2000
6300  GOSUB 3000
6400  GOSUB 4000
6500  GOSUB 5000
6600  GOSUB 5200
6700 RETURN
7000 REM ***Input partial recruitments***
7010  PRINT : CALCFLAG=1
7020  INPUT "Enter filename (including extension) of partial recruitment vector: ",FILESTORE$
7030 GOSUB 9000
7040  I8TITLE1$=TITLE1$
7050  I8TITLE2$=TITLE2$
7060  IF EFLAGI8=1 THEN ERASE I8MATRI1$
7070  EFLAGI8=1
7080  DIM I8MATRI1$ (NAGESOUT,NYEARSOUT)
7090    FOR I=1 TO NAGESOUT
7100      FOR J=1 TO NYEARSOUT
7110        I8MATRI1$(I,J)=MATRI1$(LAGE+I,J)
7120    NEXT J,I
7130  PRINT
7140  OUTFLAGI8=1
7150 RETURN
7170  TITLE1OUT$=I8TITLE1$
7180  TITLE2OUT$=I8TITLE2$
7190  ERASE MATRIIOUT$
7200  DIM MATRIXOUT$ (NAGESOUT,1)
7210    FOR I=1 TO NAGESOUT
7230      MATRIIOUT$(I,1)=I8MATRI1$(I,1)
7240    NEXT I
7250 RETURN
8000 REM ***Input independent abundance index***
8010  CLS : VWFLAG=0
8020  INPUT "Enter filename (including extension) of abundance index:   ",FILESTORE$
8030 GOSUB 9000
8035    IF NAGES=1 THEN GOSUB 9750 : VWFLAG=1
8040  PLTILABL$=TITLE1$
8050  I9TITLE2$=TITLE2$
8060  IF EFLAGI9=1 THEN ERASE PLOTI$
8070  EFLAGI9=1
8080  DIM PLOTI$ (NAGESOUT,NYEARSOUT)
8090    FOR I=1 TO NAGESOUT
8100      FOR J=1 TO NYEARSOUT
8110        IF VWFLAG=1 THEN PLOTI$(I,J)=MATRII$(I,LYEAR+J) ELSE PLOTI$(I,J)=MATRII$(LAGE+I,LYEAR+J)
8120    NEXT J,I
8130  PRINT
8140  PRINT "Strike any key to continue."
8150  A$=INKEY$ : IF A$="" THEN 8150
```

```
8155   IF VWFLAG=1 THEN GOSUB 9700
8160  OUTFLAG19=1
8170  TITLE1OUT$=PLTILABL$
8180  TITLE2OUT$=I9TITLE2$
8190  ERASE MATRIXOUT$
8191    IF VWFLAG=1 THEN 8192 ELSE 8200
8192  DIM MATRIXOUT$(1,NYEARSOUT)
8193    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=PLOTI$(1,J) : NEXT J
8194 RETURN
8200  DIM MATRIXOUT$ (NAGESOUT, NYEARSOUT)
8210    FOR I=1 TO NAGESOUT
8220      FOR J=1 TO NYEARSOUT
8230        MATRIXOUT$(I,J)=PLOTI$(I,J)
8240    NEXT J,I
8250 RETURN
9000 REM ***Diskette input***
9040  INFILESPEC$="B:"+FILESTORE$
9045    ON ERROR GOTO 9300
9050    OPEN INFILESPEC$ FOR INPUT AS #1
9060      LINE INPUT#1, TITLE1$
9070      LINE INPUT#1, TITLE2$
9080      LINE INPUT#1, PRECISION$
9090      INPUT#1, NAGES,NYEARS,FAGE,FYEAR
9100      IF ERASEFLAG=1 THEN ERASE MATRI$
9110      ERASEFLAG=1
9120      DIM MATRI$ (NAGES,NYEARS)
9130        FOR I=1 TO NAGES
9140          FOR J=1 TO NYEARS
9150            INPUT#1, MATRI$(I,J)
9160        NEXT J,I
9170    CLOSE#1 : ON ERROR GOTO 0
9180  PRINT
9190    IF (PRECISION$="D") OR (PRECISION$="d") THEN PRINT "Data was stored in double precision." ELSE PRINT
       "Data was stored in single precision."
9220  DISKFLAG=1
9230 RETURN
9300 REM ***Trap 'File not found' error***
9310    IF ERR=53 THEN PRINT CHR$(7) : PRINT "File not found. Probable cause: wrong diskette and/or filename." : INPUT
       "Re-enter filename:   ",FILESTORE$ : RESUME 9040
9320    ON ERROR GOTO 0
9500 REM ***List files on drive B***
9510  CLS
9520  FILES "B:*.*"
9530  PRINT
9540  PRINT "Strike any key to continue."
9550  A$=INKEY$ : IF A$="" THEN 9550
9560 RETURN
9700 REM ***Temporary storage (vector by ages)***
9705  TEMPNYEARSOUT=NYEARSOUT
9710  TEMPFYEAROUT=FYEAROUT
9715  FYEAROUT=FYEAROUT+NYEARSOUT-1
9720  NYEARSOUT=1
9725 RETURN
9730 REM ***Reassign temporary variables (ages)***
9735  NYEARSOUT=TEMPNYEARSOUT
9740  FYEAROUT=TEMPFYEAROUT
9745 RETURN
9750 REM ***Temporary storage (vector by years)***
9755  TEMPNAGESOUT=NAGESOUT
9760  TEMPFAGEOUT=FAGEOUT
9765  FAGEOUT=FAGEOUT+NAGESOUT-1
9770  NAGESOUT=1
9775 RETURN
9780 REM ***Reassign temporary variables (years)***
9785  NAGESOUT=TEMPNAGESOUT
9790  FAGEOUT=TEMPFAGEOUT
9795 RETURN
```

```
10000 REM ***Subroutines for calculation menu item output***
13060  TITLE1OUT$=C1TITLE1$
13070  TITLE2OUT$=C1TITLE2$
13080  ERASE MATRIXOUT$
13090  DIM MATRIXOUT$ (NAGESOUT,NYEARSOUT)
13100    FOR I=1 TO NAGESOUT
13110      FOR J=1 TO NYEARSOUT
13120        MATRIXOUT$(I,J)=C1MATRIX$(I,J)
13130    NEXT J,I
13140 RETURN
13260  TITLE1OUT$=C2TITLE1$
13270  TITLE2OUT$=C2TITLE2$
13280  ERASE MATRIXOUT$
13290  DIM MATRIXOUT$ (NAGESOUT,NYEARSOUT)
13300    FOR I=1 TO NAGESOUT
13310      FOR J=1 TO NYEARSOUT
13320        MATRIXOUT$(I,J)=C2MATRIX$(I,J)
13330    NEXT J,I
13340 RETURN
13460  TITLE1OUT$=C3TITLE1$
13470  TITLE2OUT$=C3TITLE2$
13480  ERASE MATRIXOUT$
13490  DIM MATRIXOUT$ (NAGESOUT,NYEARSOUT)
13500    FOR I=1 TO NAGESOUT
13510      FOR J=1 TO NYEARSOUT
13520        MATRIXOUT$(I,J)=C3MATRIX$(I,J)
13530    NEXT J,I
13540 RETURN
13660  TITLE1OUT$=C4TITLE1$
13670  TITLE2OUT$=C4TITLE2$
13680  ERASE MATRIXOUT$
13690  DIM MATRIXOUT$ (NAGESOUT,NYEARSOUT)
13700    FOR I=1 TO NAGESOUT
13710      FOR J=1 TO NYEARSOUT
13720        MATRIXOUT$(I,J)=C4MATRIX$(I,J)
13730    NEXT J,I
13740 RETURN
14240  TITLE1OUT$=C5TITLE1$
14250  TITLE2OUT$=C5TITLE2$
14260  ERASE MATRIXOUT$
14270  DIM MATRIXOUT$ (1,NYEARSOUT)
14280    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=C5MATRIX$(1,J) : NEXT
14290 RETURN
15250  TITLE1OUT$=C6TITLE1$
15260  TITLE2OUT$=C6TITLE2$
15270  ERASE MATRIXOUT$
15280  DIM MATRIXOUT$ (1,NYEARSOUT)
15290    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=C6MATRIX$(1,J) : NEXT
15300 RETURN
16260  TITLE1OUT$=C7TITLE1$
16270  TITLE2OUT$=C7TITLE2$
16280  ERASE MATRIXOUT$
16290  DIM MATRIXOUT$ (1,NYEARSOUT)
16300    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=C7MATRIX$(1,J) : NEXT
16310 RETURN
17380  TITLE1OUT$=C8TITLE1$
17390  TITLE2OUT$=C8TITLE2$
17400  ERASE MATRIXOUT$
17410  DIM MATRIXOUT$ (1,NYEARSOUT)
17420    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=C8MATRIX$(1,J) : NEXT
17430 RETURN
18270  TITLE1OUT$=C9TITLE1$
18280  TITLE2OUT$=C9TITLE2$
18290  ERASE MATRIXOUT$
18300  DIM MATRIXOUT$ (1,NYEARSOUT)
18310    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=C9MATRIX$(1,J) : NEXT
18320 RETURN
```

```
19320 TITLE1OUT$=C1OTITLE1$
19330 TITLE2OUT$=C1OTITLE2$
19340 ERASE MATRIXOUT$
19350 DIM MATRIXOUT$ (NAGESOUT,NYEARSOUT)
19360   FOR I=1 TO NAGESOUT
19370     FOR J=1 TO NYEARSOUT
19380       MATRIXOUT$(I,J)=C1OMATRIX$(I,J)
19390   NEXT J,I
19400 RETURN
19690 TITLE1OUT$=C1OTITLE1$
19700 TITLE2OUT$=C1OTITLE2$
19710 ERASE MATRIXOUT$
19720 DIM MATRIXOUT$ (NAGESOUT,1)
19730   FOR I=1 TO NAGESOUT
19740     MATRIXOUT$(I,1)=C1OMATRIX$(I,1)
19750   NEXT I
19760 RETURN
20000 REM ***OUTPUT***
20010 CLS
20020 PRINT "        OUTPUT MENU FOR "SHORT ANALYSIS "; : COLOR 7,0 : PRINT "(" NOWOUT$ ")" : COLOR 2,0
20030 PRINT
20040 PRINT "F1. Print on screen"
20050 PRINT "F2. Print on line printer (will print filename if F3 was last item chosen"
20055 PRINT "    and output menu was invoked from a calculation menu item)"
20060 PRINT "F3. Output on diskette"
20065 PRINT "F5. Go to data ENTRY program"
20095 PRINT "F6. END program"
20100 PRINT "CTRL F1. INPUT MENU"
20110 PRINT "CTRL F2. CALCULATION MENU"
20120 RETURN
20200 REM ***From menu***
20205 COLOR 2,0 : NOWOUT$="GENERAL"
20210 GOSUB 20000
20220 C$=INKEY$ : IF C$="" THEN 20220
20230 CHECK1=ASC(C$)
20240 LOCATE 24,1,0 : PRINT SPACE$(79);
20250   IF CHECK1<>0 THEN LOCATE 24,1,0 : PRINT CHR$(7) "Please press a valid F-key."; : GOTO 20220
20260 EXITCODE=ASC(MID$(C$,2))
20270   ON EXITCODE-58 GOSUB 22000,24000,26000
20275   IF (EXITCODE-58<4) THEN 20200
20290   IF EXITCODE=63  THEN CLS : PRINT "Chaining to ENTRY program..." : CLEAR : CHAIN "ENTRY"
20300   IF EXITCODE=64  GOTO  29000
20310   IF EXITCODE=94  THEN VAFLAG=0 : VYFLAG=0 : GOTO 100
20320   IF (EXITCODE=93) AND (CALCFLAG=1) THEN VAFLAG=0 : VYFLAG=0 : CHAIN "cohortc",10000,ALL
20330   IF (EXITCODE=93) AND (CALCFLAG=0) THEN VAFLAG=0 : VYFLAG=0 : CHAIN "cohortc",10600,ALL
20340 LOCATE 24,1,0 : PRINT CHR$(7) "Please press a valid F-key."; : GOTO 2022
21000 REM ***Items available for printing***
21010   IF OUTFLAG11 =1 THEN LOCATE 14,1 ,0 : OUTVECI(1)=1 : PRINT "11.  Catch at age matrix"
21020   IF OUTFLAG12 =1 THEN LOCATE 15,1 ,0 : OUTVECI(2)=1 : PRINT "12.  Weight at age matrix"
21030   IF OUTFLAG13 =1 THEN LOCATE 16,1 ,0 : OUTVECI(3)=1 : PRINT "13.  Fishing mortality, last year"
21040   IF OUTFLAG14 =1 THEN LOCATE 17,1 ,0 : OUTVECI(4)=1 : PRINT "14.  Fishing mortality, last age"
21050   IF OUTFLAG15 =1 THEN LOCATE 18,1 ,0 : OUTVECI(5)=1 : PRINT "15.  Natural mortality"
21060   IF OUTFLAG16 =1 THEN LOCATE 19,1 ,0 : OUTVECI(6)=1 : PRINT "16.  Partial recruitment (input)"
21070   IF OUTFLAG19 =1 THEN LOCATE 20,1 ,0 : OUTVECI(7)=1 : PRINT "17.  Independent index"
21075 LOCATE 24,1,0 : PRINT "IA.  Print all input items";
21080   IF OUTFLAGC1 =1 THEN LOCATE 14,40,0 : OUTVECI(8)=1 : PRINT "C1.  Numbers at age, beginning of year"
21090   IF OUTFLAGC2 =1 THEN LOCATE 15,40,0 : OUTVECI(9)=1 : PRINT "C2.  Numbers at age, midyear"
21100   IF OUTFLAGC3 =1 THEN LOCATE 16,40,0 : OUTVECI(10)=1 : PRINT "C3.  Biomass, midyear or fishable"
21110   IF OUTFLAGC4 =1 THEN LOCATE 17,40,0 : OUTVECI(11)=1 : PRINT "C4.  Age specific fishing mortalities"
21120   IF OUTFLAGC5 =1 THEN LOCATE 18,40,0 : OUTVECI(12)=1 : PRINT "C5.  Total numbers, beginning of year"
21130   IF OUTFLAGC6 =1 THEN LOCATE 19,40,0 : OUTVECI(13)=1 : PRINT "C6.  Total numbers, midyear"
21140   IF OUTFLAGC7 =1 THEN LOCATE 20,40,0 : OUTVECI(14)=1 : PRINT "C7.  Total biomass, midyear or fishable"
21150   IF OUTFLAGC8 =1 THEN LOCATE 21,40,0 : OUTVECI(15)=1 : PRINT "C8.  Mean fishing mortalities"
21160   IF OUTFLAGC9 =1 THEN LOCATE 22,40,0 : OUTVECI(16)=1 : PRINT "C9.  Overall fishing mortalities"
21170   IF OUTFLAGC10=1 THEN LOCATE 23,40,0 : OUTVECI(17)=1 : PRINT "C10. Partial recruitment (calc'd)"
21175 LOCATE 24,40,0 : PRINT "CA.  Print all calculation items";
```

```
21180   IF EXITCODE=61 THEN LOCATE 18,1 : PRINT SPACE$(30);
21190   IF (EXITCODE=59) OR (EXITCODE=61) THEN LOCATE 24,1,0 : PRINT SPACE$(75);
21200 LOCATE 11,1 : INPUT "Enter code for item to be output:   ",ITEMCODE$
21210   IF (ITEMCODE$="ia") OR (ITEMCODE$="IA") THEN CLS : ALLOUTFLAG=1 : GOSUB 21800 : RETURN
21215   IF (ITEMCODE$="ca") OR (ITEMCODE$="CA") THEN CLS : ALLOUTFLAG=1 : GOSUB 21900 : RETURN
21220 WHATOUTI=VAL(MID$(ITEMCODE$,2))
21225   IF (LEFT$(ITEMCODE$,1)="I") OR (LEFT$(ITEMCODE$,1)="i") THEN GOSUB 21240 : RETURN
21230   IF (LEFT$(ITEMCODE$,1)="C") OR (LEFT$(ITEMCODE$,1)="c") THEN GOSUB 21270 : RETURN
21240     ON WHATOUTI GOSUB 21260,21260, 9700, 9750,21260, 9700,21300
21245     ON WHATOUTI GOSUB  1360, 2095, 3930, 4750,21260, 7170, 8170
21250     ON WHATOUTI GOSUB 21500,21500,21500,21500,21700,21500,21500
21255     ON WHATOUTI GOSUB 21260,21260, 9730, 9780,21260, 9730,21310
21260   RETURN
21270     ON WHATOUTI GOSUB 21290,21290,21290,21290, 9750, 9750, 9750, 9750, 9750,21320
21275     ON WHATOUTI GOSUB 13060,13260,13460,13660,14240,15250,16260,17380,18270,21330
21280     GOSUB 21500
21285     ON WHATOUTI GOSUB 21290,21290,21290,21290, 9780, 9780, 9780, 9780, 9780,21340
21290   RETURN
21300     IF VNFLAG=1 THEN GOSUB 9750
21305   RETURN
21310     IF VNFLAG=1 THEN GOSUB 9780
21315   RETURN
21320     IF PRFLAG=1 THEN GOSUB 9700
21325   RETURN
21330     IF PRFLAG=1 THEN GOSUB 19690 ELSE GOSUB 19320
21335   RETURN
21340     IF PRFLAG=1 THEN GOSUB 9730
21345   RETURN
21380 BEEP : LOCATE 11,1 : PRINT SPACE$(79); : GOTO 21200
21500 REM ***Screen, printer, or diskette output***
21510   ON EXITCODE-58 GOSUB 22900,24900,27000 : RETURN
21700 REM ***Output natural mortality***
21710   IF EXITCODE=59 THEN GOSUB 22300 : RETURN
21720   IF EXITCODE=60 THEN GOSUB 24300 : RETURN
21730   IF EXITCODE=61 THEN RETURN 21380
21800 REM ***Print all available input menu items***
21810   FOR NOWPRINT=1 TO 7
21820     IF OUTVECI(NOWPRINT)=1 THEN WHATOUTI=NOWPRINT : GOSUB 21240
21830   NEXT
21840 ALLOUTFLAG=0 : DS$="" : RETURN
21900 REM ***Print all available calculation menu items***
21910   FOR NOWPRINT=8 TO 17
21920     IF OUTVECI(NOWPRINT)=1 THEN WHATOUTI=NOWPRINT-7 : GOSUB 21270
21930   NEXT
21940 ALLOUTFLAG=0 : DS$="" : RETURN
22000 REM ***Print on screen***
22010 CLS : VAFLAG=0 : VYFLAG=0 : DISKFLAG=0
22020 PRINT "The following is available for screen display. The numbering is almost"
22030 PRINT "the same as in the input menu (I) and calculation menu (C). To select"
22040 PRINT "an item for printing, enter the code preceding the item (eg. CA).
22050 PRINT "Matrices with <=8 years of data will be printed on one screen, "
22060 PRINT "but matrices with >8 years will need at least two screens. To "
22070 PRINT "print the next screen, press 'Pg Up'. To reprint the previous"
22080 PRINT "screen, press 'Pg Dn'. Press 'I' to exit to output menu. Each item"
22090 PRINT "is printed with a title, ages and years (if applicable)."
22100 GOSUB 21000 : RETURN
22300 REM ***Print natural mortality***
22310 CLS
22320 PRINT "Natural mortality is :   ";NMORT
22330 PRINT
22340 PRINT "Press any key to return to output menu."
22350 C$=INKEY$ : IF C$="" THEN 22350
22360 RETURN
22900 LOCATE 12,1
23000 REM ***Display data***
23010 LOCATE 12,1 : PRINT SPACE$(79); : LOCATE 12,1
23020 PRINT "Enter desired number of decimal places for " TITLE1OUT$                " (0-4):   "; : DISKFLAG=0
```

```
23030  DECIOUT$=INKEY$ : IF DECIOUT$="" THEN 23030
23040  PRINT DECIOUT$ : DECIOUT=VAL(DECIOUT$)
23050    IF DECIOUT=0 THEN SOFTPR$="##########" : GOTO 23070
23060    IF (DECIOUT>0) AND (DECIOUT<=4) THEN SOFTPR$=STRING$(8-DECIOUT,"#")+"."+STRING$(DECIOUT,"#") ELSE GOTO 23010
23070  PAGE=1 : OUTFLAG=0
23080  PAGES=CINT((NYEARSOUT/8)+.45)
23090    ON PAGE GOSUB 23760,23800,23840 : GOSUB 23130 : GOTO 23120
23120    IF OUTFLAG=1 THEN RETURN ELSE GOTO 23090
23130 REM ***Print matrix***
23140  LYR=LY+FYEAROUT-1
23150  HYR=HY+FYEAROUT-1
23160  COLOR 2,0
23170  CLS
23180  COLOR 1,7
23190  PRINT SPACE$(79);
23200  LOCATE 1,1,0 : PRINT TITLE1OUT$; : IF (TITLE2OUT$=" ") OR (TITLE2OUT$="") THEN 23210 ELSE PRINT " : " LEFT$(TITLE2OUT$,70);
       " (RUN #";STR$(RUNNUM);")"
23210  COLOR 7,1
23220  LOCATE 2,1 : PRINT "  YEAR "
23230    FOR I=LY TO HY
23240      DUMMY=I
23250        IF LY=; THEN DUMMY=DUMMY-8
23260        IF LY=17 THEN DUMMY=DUMMY-16
23270      PY=DUMMY*9-1
23280      PYR=FYEAROUT+I-1
23290      LOCATE 2,PY : PRINT USING "##########";PYR
23300    NEXT
23310  PRINT
23320    IF NAGESOUT=1 THEN PRINT "       " ELSE PRINT "  AGE  "
23330    FOR J=1 TO NAGESOUT
23340      IAGE=FAGEOUT+J-1
23350        IF NAGESOUT=1 THEN PRINT "    "; ELSE PRINT USING "####";IAGE;
23360        IF J=NAGESOUT THEN PRINT "   "; ELSE PRINT "   "
23370    NEXT J
23380    FOR K=1 TO NAGESOUT
23390      ROWE=K+4
23400      LOCATE ROWE,8
23410        FOR L=LY TO HY
23420          EVENODD=(INT(L/2))*2
23430            IF L=EVENODD THEN COLOR 0,2 ELSE COLOR 2,0
23440          PRINT USING SOFTPR$;MATRIXOUT$(K,L);
23450    NEXT L,K
23460 REM ***Change screen***
23470  PRINTING=-1
23480    IF (PAGE<PAGES) AND (LY=9) THEN HY=8
23490    IF (PAGE=PAGES) AND (LY=9) THEN HY=NYEARS-8
23500    IF (PAGE=PAGES) AND (LY=17) THEN HY=NYEARS-16
23510    WHILE PRINTING
23520      ED$=INKEY$ : IF ED$="" THEN 23520
23530        IF (ED$="I") OR (ED$="i") THEN GOSUB 23590 : GOTO 23570
23540      EDCODE=ASC(ED$)
23550        IF EDCODE=0 THEN GOSUB 23610 : GOTO 23570     'ED$=extended code
23560      BEEP
23570    WEND
23580 RETURN
23590 REM ***Finish Printing***
23600  PRINTING=0 : OUTFLAG=1 : RETURN
23610 REM ***Extended code***
23620  EXCODE=ASC(MID$(ED$,2))
23630  IF EXCODE=73  THEN GOSUB 23660 : RETURN      'Pg Up
23640  IF EXCODE=81  THEN GOSUB 23690 : RETURN      'Pg Dn
23650  BEEP : RETURN
23660 REM ***Page up***
23670  IF PAGE=PAGES THEN BEEP : RETURN
23680  PAGE=PAGE+1 : PRINTING=0 : RETURN
```

```
23690 REM ***Page down***
23700  IF PAGE=1 THEN BEEP : RETURN
23710  PAGE=PAGE-1 : PRINTING=0 : RETURN
23760 REM ***Page 1***
23770  LY=1
23780  IF PAGE<PAGES THEN HY=8 ELSE HY=NYEARSOUT
23790  RETURN
23800 REM ***Page 2***
23810  LY=9
23820  IF PAGE<PAGES THEN HY=16 ELSE HY=NYEARSOUT
23830  RETURN
23840 REM ***Page 3***
23850  LY=17
23860  HY=NYEARSOUT
23870  RETURN
24000 REM ***Hardcopy***
24010  CLS : VAFLAG=0 : VYFLAG=0
24020  PRINT "The following is available for hardcopy. The numbering is almost the"
24030  PRINT "same as in the input menu (I) and calculation menu (C). To select
24040  PRINT "an item for printing, enter the code preceding the item (eg. CA).
24050  PRINT "Each item is output with a title, ages and years (if applicable).
24060  PRINT "When using IA or CA option, default number of decimal places is"
24070  PRINT "zero for any population numbers or biomass options; three for all
24080  PRINT "other output."
24100 GOSUB 21000 : RETURN
24300 REM ***Print natural mortality***
24310  LPRINT : LPRINT : LPRINT
24320  LPRINT "Natural mortality is :   "!MMORT
24330 RETURN
24900  LOCATE 12,1
25000 REM ***Hardcopy sub-routine***
25020  LPRINT
25030  LPRINT
25040    IF DISKFLAG=1 THEN LPRINT CHR$(18) "Name of diskette file is:   "; FILESTORE$ : DISKFLAG=0 : FILESTORE$="" : LPRINT
25041    IF ALLOUTFLAG=1 GOTO 25042 ELSE GOTO 25050
25042    IF DS$="" GOTO 25043 ELSE GOTO 25046
25043  PRINT "Default number of decimal places or Specify? (D/S):   ";
25044  DS$=INKEY$ : IF DS$="" THEN 25044
25045  PRINT DS$
25046    IF (DS$="D") OR (DS$="d") THEN DECIOUT=DEFDECI(NOWPRINT) : GOTO 25090
25047    IF (DS$="S") OR (DS$="s") THEN 25049 ELSE LOCATE 12,1 : PRINT SPACE$(79); : LOCATE 12,1 : GOTO 25043
25049  LOCATE 12,1 : PRINT SPACE$(79); : LOCATE 12,1
25050  PRINT "Enter desired number of decimal places for " TITLE1OUT$ " (0-4):   ";
25060  DECIOUT$=INKEY$ : IF DECIOUT$="" THEN 25060
25070  PRINT DECIOUT$ : DECIOUT=VAL(DECIOUT$)
25080    IF (DECIOUT<0) OR (DECIOUT>4) GOTO 25049
25090  MAXTESTOUT$=-999999999$
25100    FOR I=1 TO NAGESOUT
25110      FOR J=1 TO NYEARSOUT
25120        IF ABS(MATRIXOUT$(I,J))>MAXTESTOUT$ THEN MAXTESTOUT$=ABS(MATRIXOUT$(I,J))
25130    NEXT J,I
25140  MAXOUT=LEN(STR$(INT(ABS(MAXTESTOUT$))))-1
25150    IF DECIOUT=0 THEN MAXLENOUT=MAXOUT+2 : GOTO 25170
25160  MAXLENOUT=MAXOUT+DECIOUT+3
25170    IF MAXLENOUT<5 THEN MAXLENOUT=5
25180  NFIELDSOUT=INT(126/MAXLENOUT)
25190    IF DECIOUT=0 THEN HARDPROUT$=STRING$(MAXLENOUT,"0") : GOTO 25210
25200  HARDPROUT$=STRING$(MAXOUT+2,"0")+"."+STRING$(DECIOUT,"0")
25210  YEARPROUT$=STRING$(MAXLENOUT,"0")
25220    IF NYEARSOUT>NFIELDSOUT THEN BLOCKSOUT=2 ELSE BLOCKSOUT=1
25230  WIDTH "LPT1:",132
25240    IF BLOCKSOUT=1 THEN LYOUT=1 : HYOUT=NYEARSOUT : GOSUB 25260 : WIDTH "LPT1:",80 : RETURN
25250    IF BLOCKSOUT=2 THEN GOSUB 25500 : WIDTH "LPT1:",80 : RETURN
25260 REM ***Print data***
25270  LPRINT CHR$(15); TITLE1OUT$; : IF (TITLE2OUT$=" ") OR (TITLE2OUT$="") THEN LPRINT TAB(120) DATE$ ELSE LPRINT " : " TITLE2OUT$
       " (RUN #" STR$(RUNNUM) ")" TAB(120) DATE$
```

```
25280  LPRINT
25290  LPRINT
25300  LPRINT CHR$(15); "YEAR ";
25310    FOR I=LYOUT TO HYOUT
25320      YEAROUT=FYEAROUT+I-1
25330      LPRINT USING YEARPROUT$;YEAROUT;
25340    NEXT I
25350  LPRINT
25360  LPRINT
25370    IF NAGESOUT=1 THEN LPRINT "     " ELSE LPRINT "AGE  "
25380  LPRINT
25390    FOR K=1 TO NAGESOUT
25400      AGEOUT=FAGEOUT+K-1
25410        IF NAGESOUT=1 THEN LPRINT "  "; ELSE LPRINT USING "##";AGEOUT;
25420      LPRINT "   ";
25430        FOR L=LYOUT TO HYOUT
25440          LPRINT USING HARDPROUT$;MATRIIOUT$(K,L);
25450        NEXT L
25460      LPRINT
25470    NEXT K
25480  LPRINT CHR$(18)
25490 RETURN
25500 REM ***Print data in two or more blocks***
25510  LYOUT=1
25520  HYOUT=NFIELDSOUT
25530 GOSUB 25260
25540  LYOUT=LYOUT+NFIELDSOUT
25550  HYOUT=HYOUT+NFIELDSOUT
25560    WHILE HYOUT<NYEARSOUT
25570      GOSUB 25280
25580      LYOUT=LYOUT+NFIELDSOUT
25590      HYOUT=HYOUT+NFIELDSOUT
25600    WEND
25610  HYOUT=NYEARSOUT
25620 GOSUB 25280
25630 RETURN
26000 REM ***Output to disk***
26010  CLS : VAFLAG=0 : VYFLAG=0
26020  PRINT "The following is available for diskette output. The numbering is almost"
26030  PRINT "the same as in the input menu (I) and calculation menu (C). To select"
26040  PRINT "an item for output, enter the code preceding the item (eg. C&)."
26050  PRINT "Each item is output with a title, precision, number of ages,"
26060  PRINT "number of years, first age and first year."
26100 GOSUB 21000 : RETURN
27000 REM ***Disk output***
27010  CLS
27020  PRINT "Would you like single or double precision for diskette output? (S/D)   ";
27030  PRECISION$=INKEY$ : IF PRECISION$="" THEN 27030
27040  PRINT PRECISION$
27050    IF (PRECISION$="S") OR (PRECISION$="s") OR (PRECISION$="D") OR (PRECISION$="d") THEN 27060 ELSE GOTO 27020
27060    IF (PRECISION$="S") OR (PRECISION$="s") THEN GOSUB 27920
27070 GOSUB 27850
27080  INPUT "Enter filename:   ",FILENAME$
27090  PRINT "Insert data diskette into drive B."
27100  PRINT "Press any key to begin output to diskette."
27110  A$=INKEY$ : IF A$="" THEN 27110
27120  DOT=INSTR(FILENAME$,".")
27130    IF DOT=0 THEN FILESPEC$="B:"+FILENAME$+".DAT" ELSE FILESPEC$="B:"+FILENAME$
27140  ERRFLAG=1
27150 GOSUB 27350
27160    IF ERRFLAG=1 THEN GOSUB 27350
27170    OPEN FILESPEC$ FOR  OUTPUT AS #1
27180      PRINT#1, TITLE1OUT$
27190      PRINT#1, TITLE2OUT$
27200      PRINT#1, PRECISION$
27210      PRINT#1, NAGESOUT;NYEARSOUT;FAGEOUT;FYEAROUT;
```

```
27220       FOR I=1 TO NAGESOUT
27230         FOR J=1 TO NYEARSOUT
27240           IF (PRECISION$="D") OR (PRECISION$="d") THEN PRINT#1, MATRIXOUT#(I,J); ELSE TEMP!=MATRIXOUT#(I,J) : PRINT#1,TEMP!;
27250         NEXT J
27260       NEXT I
27270     CLOSE #1
27280     ON ERROR GOTO 0
27290   FILESTORE$=MID$(FILESPEC$,3)
27300   PRINT "Data has been stored on drive B as:   "FILESTORE$
27310   PRINT "Press any key to return to menu."
27320   DISKFLAG=1
27330   A$=INKEY$ : IF A$="" THEN 27330
27340 RETURN
27350 REM ***Check if file exists***
27360     ON ERROR GOTO 27450
27370   ERRFLAG=0
27380   LN=CSRLIN
27390   LOCATE 23,1 : COLOR 0,0
27400   FILECHECK=0
27410   FILES FILESPEC$ : FILECHECK=1
27420     IF FILECHECK=1 GOTO 27480    'File already exists
27430   COLOR 2,0 : LOCATE LN+1,1
27440     ON ERROR GOTO 27770 : RETURN
27450     IF ERR=53 THEN RESUME 27430
27460   COLOR 2,0 : LOCATE LN+1,1
27470     ON ERROR GOTO 0
27480 ON ERROR GOTO 0 : COLOR 2,0 : LOCATE LN+1,1
27490   ERRFLAG=1
27500   CLS
27510   PRINT CHR$(7) "File already exists. Would you like to: "
27520   PRINT "  A. Enter a different filename?"
27530   PRINT "  B. Rename the existing file without changing the name you have just entered?"
27540   PRINT "  C. Write over the existing file?"
27550   PRINT
27560   PRINT "Press A, B, or C on the keyboard.   ";
27570   OPT$=INKEY$ : IF OPT$="" THEN 27570
27580   PRINT OPT$
27590     IF (OPT$="A") OR (OPT$="a") THEN GOSUB 27620 : GOTO 27760
27600     IF (OPT$="B") OR (OPT$="b") THEN GOSUB 27660 : GOTO 27760
27610     IF (OPT$="C") OR (OPT$="c") THEN GOSUB 27740 : GOTO 27760 ELSE GOTO 27560
27620 INPUT "Enter new filename:   ",FILENAME$     '***Option A***
27630   DOT=INSTR(FILENAME$,".")
27640     IF DOT=0 THEN FILESPEC$="B:"+FILENAME$+".DAT" ELSE FILESPEC$="B:"+FILENAME$
27650 RETURN
27660 INPUT "Enter new filename for existing file:   ",NEWFN$   '***Option B***
27670   DOT=INSTR(NEWFN$,".")
27680     IF DOT=0 THEN NEWFS$="B:"+NEWFN$+".DAT" ELSE NEWFS$="B:"+NEWFN$
27690     ON ERROR GOTO 27820
27700   NAME FILESPEC$ AS NEWFS$
27710   NEWFS$=MID$(NEWFS$,3)
27720   PRINT "New name for existing file is:   "NEWFS$
27730 RETURN
27740 KILL FILESPEC$           '***Option C***
27750   RETURN
27760   ON ERROR GOTO 27770 : RETURN
27770 REM ***Errors for full diskette***
27780   CLS
27790     IF (ERR=61) THEN PRINT CHR$(7) "Data diskette is full." : PRINT "Insert a new diskette into drive B." : KILL FILESPEC$ :
        RESUME 27810
27800     IF (ERR=67) THEN PRINT CHR$(7) "Data diskette has too many files." : PRINT "Insert a new diskette into drive B." :
        RESUME 27810
27810     ON ERROR GOTO 0 : GOTO 27100
27820 REM ***NAME statement error***
27830     IF ERR=58 THEN PRINT CHR$(7) "New filename already exists." : RESUME 27660
27840     ON ERROR GOTO 0
```

```
27850 REM ***Instructions***
27860  PRINT
27870  PRINT "Rules for naming of files are as described in the BASIC reference
27880  PRINT "manual. If no filename extension is included then the extension "
27890  PRINT "'.dat ' is added by default."
27900  PRINT
27910 RETURN
27920 REM ***Precision verification***
27930  PRINT CHR$(7)
27940  PRINT "WARNING: Single precision output will only store 7 significant digits."
27950  PRINT "Any numbers with more than 7 significant digits will be rounded off."
27960  PRINT "Press 'D ' if you would rather have double precision, if not,"
27970  PRINT "press RETURN." : PRINT
27980  PV$=INKEY$ : IF PV$="" THEN 27980
27990    IF (PV$="D") OR (PV$="d") THEN PRECISION$="D" : RETURN
27995    IF (PV$=CHR$(13)) THEN RETURN ELSE GOTO 27960
28000 REM ***Chain to PLOT & FIT***
28010  CLS : DISKFLAG=0
28020    IF OUTFLAG19=0 THEN PRINT CHR$(7) "Independent index is not defined. Returning to input menu...." : FOR I=1 TO 3000 : NEXT
       ELSE CHAIN"preplot",,ALL
28030    IF VAFLAG=1 THEN GOSUB 9730 : VAFLAG=0
28040    IF VYFLAG=1 THEN GOSUB 9780 : VYFLAG=0
28050 GOTO 100
29000 REM ***Reset F-keys and end program***
29010  KEY 1,"list "
29020  KEY 2,"run"+CHR$(13)
29030  KEY 3,"load"+CHR$(34)
29040  KEY 4,"save"+CHR$(34)
29050  KEY 5,"cont"+CHR$(13)
29060  KEY 6,"edit "
29070  CLS
29080  KEY ON
29090 END
29200 REM ***Reset dimensions after plot program***
29210    IF VAFLAG=1 THEN GOSUB 9730 : VAFLAG=0
29220    IF VYFLAG=1 THEN GOSUB 9780 : VYFLAG=0
29230 GOTO 30
```

APPENDIX 6. Listing for COHORTC.BAS (968 lines, 34816 bytes without indentation).

```
5 '*****COHORT ANALYSIS PROGRAM (C/O)*****PROGRAMMER: ROB KORVER*****
500 CHAIN "cohort"  'In case RUN is done with COHORTC in memory
600 REM ***Subroutine from line 10020 : F's for last age from input menu or calculate here***
610  PRINT "Use the F's for the last age group entered in the Input menu or"
620  PRINT "have them Calculated from weighted F's for each year? (I/C)   ";
630  ICFLAG=INKEY$ : IF ICFLAG="" GOTO 630
640  PRINT ICFLAG : PRINT
650    IF (ICFLAG="I") OR (ICFLAG="i") THEN RETURN
660    IF (ICFLAG="C") OR (ICFLAG="c") THEN 670 ELSE GOTO 610
670  INPUT "Enter first age to be used for weighted F's:   ",FA : INPUT "Enter last age to be used for weighted F's:   ",LA
680  FAIND=FA-FAGEOUT+1 : LAIND=LA-FAGEOUT+1
690  OUTFLAG14=1 : I4TITLE1$="F, last age" : I4TITLE2$=GENTITLE$ : VPA=1
700  IF EFLAG14=1 THEN ERASE I4MATRIX$
710  EFLAG14=1
720  DIM I4MATRIX$ (1,NYEARSOUT)
730 PRINT : RETURN
1000 REM ***Subroutines for input menu item output***
1360  TITLE1OUT$=I1TITLE1$
1370  TITLE2OUT$=I1TITLE2$
1380  ERASE MATRIXOUT$
1390  DIM  MATRIXOUT$ (NAGESOUT, NYEARSOUT)
1400    FOR I=1 TO NAGESOUT
1410      FOR J=1 TO NYEARSOUT
1420        MATRIXOUT$(I,J)=I1MATRIX$(I,J)
1430    NEXT J,I
1440 RETURN
2095  TITLE1OUT$=I2TITLE1$
2100  TITLE2OUT$=I2TITLE2$
2105  ERASE MATRIXOUT$
2110  DIM  MATRIXOUT$ (NAGESOUT, NYEARSOUT)
2115    FOR I=1 TO NAGESOUT
2120      FOR J=1 TO NYEARSOUT
2125        MATRIXOUT$(I,J)=I2MATRIX$(I,J)
2130    NEXT J,I
2135 RETURN
3930  TITLE1OUT$=I3TITLE1$
3940  TITLE2OUT$=I3TITLE2$
3950  ERASE MATRIXOUT$
3960  DIM MATRIXOUT$(NAGESOUT,1)
3970    FOR I=1 TO NAGESOUT : MATRIXOUT$(I,1)=I3MATRIX$(I,1) : NEXT
3980 RETURN
4750  TITLE1OUT$=I4TITLE1$
4760  TITLE2OUT$=I4TITLE2$
4770  ERASE MATRIXOUT$
4780  DIM MATRIXOUT$(1,NYEARSOUT)
4790    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=I4MATRIX$(1,J) : NEXT
4800 RETURN
7170  TITLE1OUT$=I8TITLE1$
7180  TITLE2OUT$=I8TITLE2$
7190  ERASE MATRIXOUT$
7200  DIM  MATRIXOUT$ (NAGESOUT,1)
7210    FOR I=1 TO NAGESOUT
7230      MATRIXOUT$(I,1)=I8MATRIX$(I,1)
7240    NEXT I
7250 RETURN
8170  TITLE1OUT$=PLTLABL$
8180  TITLE2OUT$=I9TITLE2$
8190  ERASE MATRIXOUT$
8191    IF VMFLAG=1 THEN 8192 ELSE 8200
8192  DIM MATRIXOUT$(1,NYEARSOUT)
8193    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=PLOTI$(1,J) : NEXT J
8194 RETURN
```

```
8200  DIM  MATRIXOUT#(MAGESOUT, MYEARSOUT)
8210    FOR I=1 TO MAGESOUT
8220      FOR J=1 TO MYEARSOUT
8230        MATRIXOUT#(I,J)=PLOTI#(I,J)
8240    NEXT J,I
8250 RETURN
9700 REM ***Temporary storage (vector by ages)***
9705  TEMPMYEARSOUT=MYEARSOUT
9710  TEMPFYEAROUT=FYEAROUT
9715  FYEAROUT=FYEAROUT+MYEARSOUT-1
9720  MYEARSOUT=1
9725 RETURN
9730 REM ***Reassign temporary variables (ages)***
9735  MYEARSOUT=TEMPMYEARSOUT
9740  FYEAROUT=TEMPFYEAROUT
9745 RETURN
9750 REM ***Temporary storage (vector by years)***
9755  TEMPMAGESOUT=MAGESOUT
9760  TEMPFAGEOUT=FAGEOUT
9765  FAGEOUT=FAGEOUT+MAGESOUT-1
9770  MAGESOUT=1
9775 RETURN
9780 REM ***Reassign temporary variables (years)***
9785  MAGESOUT=TEMPMAGESOUT
9790  FAGEOUT=TEMPFAGEOUT
9795 RETURN
10000 REM ***CALCULATION***
10010  CLS : COLOR 2,0 : CALCFLAG=0 : DISKFLAG=0
10020 GOSUB 600
10030    IF (OUTFLAGI1<>1) OR (OUTFLAGI3<>1) OR (OUTFLAGI4<>1) THEN PRINT CHR$(7)
         "Not all input items are defined. Returning to input menu....." : CHAIN "cohort",100,ALL
10035  PRINT "One moment please. I am doing your cohort analysis calculations."
10040  PRINT "                                    Signed, IBMPC"
10045  IF EFLAGC1=1 THEN ERASE C1MATRI#
10050  IF EFLAGC2=1 THEN ERASE C2MATRI#
10055  IF EFLAGC3=1 THEN ERASE C3MATRI#
10060  IF EFLAGC4=1 THEN ERASE C4MATRI#
10065  EFLAGC1=1
10070  EFLAGC2=1
10075  IF OUTFLAGI2=1 THEN EFLAGC3=1
10080  EFLAGC4=1
10085  DIM C1MATRI# (MAGESOUT,MYEARSOUT)
10090  DIM C2MATRI# (MAGESOUT,MYEARSOUT)
10095  IF OUTFLAGI2=1 THEN DIM C3MATRI# (MAGESOUT,MYEARSOUT)
10100  DIM C4MATRI# (MAGESOUT,MYEARSOUT)
10105    FOR I= 1 TO MAGESOUT
10110      C1MATRI#(I,MYEARSOUT)=(I1MATRI#(I,MYEARSOUT)+(I3MATRI#(I,1)*MMORT))/(I3MATRI#(I,1)*(1-(EXP(-(I3MATRI#(I,1)
           *MMORT)))))
10115      C4MATRI#(I,MYEARSOUT)=I3MATRI#(I,1)
10120    NEXT
10130    IF (ICFLAG$="1") OR (ICFLAG$="i") THEN 10150 ELSE GOTO 10200
10150    FOR J=1 TO (MYEARSOUT-1)
10155      C1MATRI#(MAGESOUT,J)=(I1MATRI#(MAGESOUT,J)+(I4MATRI#(1,J)*MMORT))/(I4MATRI#(1,J)*(1-(EXP(-(I4MATRI#(1,J)+MMORT)))))
10160      C4MATRI#(MAGESOUT,J)=I4MATRI#(1,J)
10165    NEXT
10170    FOR I= (MAGESOUT-1) TO 1 STEP -1
10175      FOR J=(MYEARSOUT-1) TO 1 STEP -1
10180        C1MATRI#(I,J)=(I1MATRI#(I,J)+EXP(MMORT/2))+(C1MATRI#(I+1,J+1)*EXP(MMORT))
10185          IF C1MATRI#(I+1,J+1)=0 THEN C4MATRI#(I,J)=0 ELSE C4MATRI#(I,J)=LOG(C1MATRI#(I,J)/C1MATRI#(I+1,J+1))-MMORT
10190    NEXT J,I
10195 GOTO 10510
10200 REM ***Use weighted F's to calcaulate F for last year***
10210  I4MATRI#(1,MYEARSOUT)=I3MATRI#(MAGESOUT,1)
10220    FOR J=MYEARSOUT-1 TO 1 STEP -1
10230      FOR I=1 TO MAGESOUT-1
10240        C1MATRI#(I,J)=(I1MATRI#(I,J)+EXP(MMORT/2))+(C1MATRI#(I+1,J+1)*EXP(MMORT))
10250          IF C1MATRI#(I+1,J+1)=0 THEN C4MATRI#(I,J)=0 ELSE C4MATRI#(I,J)=LOG(C1MATRI#(I,J)/C1MATRI#(I+1,J+1))-MMORT
10260      NEXT I
```

```
10270     SUM1#=0 : SUM2#=0
10280     FOR I=FAIND# TO LAIND#
10290       SUM1#=SUM1#+(C1#MATRIX#(I,J)*C4#MATRIX#(I,J))
10300       SUM2#=SUM2#+C1#MATRIX#(I,J)
10310     NEXT I
10320     I4#MATRIX#(1,J)=SUM1#/SUM2#
10330     C4#MATRIX#(#AGESOUT,J)=I4#MATRIX#(1,J)
10340     C1#MATRIX#(#AGESOUT,J)=(I1#MATRIX#(#AGESOUT,J)+((I4#MATRIX#(1,J)*#SHORT))/(I4#MATRIX#(1,J)*(1-(EXP(-((I4#MATRIX#(1,J)+#SHORT)))))
10350   NEXT J
10510   FOR I=1 TO #AGESOUT
10515     FOR J=1 TO #YEARSOUT
10520       C2#MATRIX#(I,J)=C1#MATRIX#(I,J)*(EXP(-(C4#MATRIX#(I,J)+#SHORT)/2))
10525         IF OUTFLAG12=1 THEN C3#MATRIX#(I,J)=C2#MATRIX#(I,J)*I2#6TRIX#(I,J)
10527         IF C4#MATRIX#(I,J)<=0 THEN C4#MATRIX#(I,J)=.0000001
10530   NEXT J,I
10535 C1TITLE1#="N at beg. year"
10540 C2TITLE1#="N at midyear"
10545 C3TITLE1#="Midyear biomass" : MF$="N"
10550 C4TITLE1#="Age-specific F"
10555 C1TITLE2#=GENTITLE$
10560 C2TITLE2#=GENTITLE$
10565 C3TITLE2#=GENTITLE$
10570 C4TITLE2#=GENTITLE$
10575 OUTFLAGC1=1
10580 OUTFLAGC2=1
10585 IF OUTFLAG12=1 THEN OUTFLAGC3=1
10590 OUTFLAGC4=1
10600 CLS : DISKFLAG=0
11000 PRINT "         CALCULATION MENU FOR COHORT ANALYSIS"
11010 PRINT
11020 PRINT "Numbers at age (beginning of year and midyear) and age-specific"
11030 PRINT "fishing mortalities are automatically calculated for any given"
11040 PRINT "set of input data when this menu appears. If a weight at age matrix"
11050 PRINT "is present, then a midyear biomass matrix will be calculated."
11060 PRINT "All the options indicated exit to the output menu automatically."
11070 PRINT "Any action taken by the output menu will be on the variable"
11080 PRINT "calculated in that option. If the output menu is directly invoked"
11090 PRINT "from here, then a selection can be made from a list of items"
11095 PRINT "presently defined and available for output."
11100 PRINT
11110 PRINT "F1.  Numbers at age, beginning of year"
11120 PRINT "F2.  Numbers at age, midyear"
11130 PRINT "F3.  Biomass, midyear or fishable"
11140 PRINT "F4.  Age-specific fishing mortality"
11150 PRINT "F5.  Total numbers, beginning of year"
11160 PRINT "F6.  Total numbers, midyear"
11170 PRINT "F7.  Total biomass (use only if F3 is present)"
11180 PRINT "F8.  Mean fishing mortalities "
11190 PRINT "F9.  Overall fishing mortalities "
11200 PRINT "F10. Partial recruitment menu"
11210 PRINT "CTRL F1. INPUT MENU"
11220 PRINT "CTRL F3. OUTPUT MENU";
11230 C9=INKEY$ : IF C9="" THEN 11230
11240 CHECK1=ASC(C9)
11250 LOCATE 25,1,0
11260   IF CHECK1<>0 THEN PRINT CHR$(7) SPACE$(40) "Please press a valid F-key.";: : GOTO 11230
11270 EXTCODE=ASC(MID$(C9,2))
11280   ON EXTCODE-58 GOSUB 13000,13200,13400,13600,14000,15000,16000,17000,18000,19000
11290   IF (EXTCODE-58<11) THEN 20400
11300   IF EXTCODE=94  THEN CHAIN "cohort",100,ALL
11310   IF EXTCODE=96  GOTO 20200
11400 LOCATE 25,1,0 : PRINT CHR$(7) SPACE$(40) "Please press a valid F-key.";: : GOTO 11230
13000 REM ***Numbers at age, beginning of year***
13010 CLS
13020 PRINT "Estimates for population numbers at age at the beginning of the year"
13030 PRINT "have been calculated."
```

```
13040  PRINT "Press any key to display output menu."
13050  A$=INKEY$ : IF A$="" GOTO 13050
13060  TITLE1OUT$=C1TITLE1$
13070  TITLE2OUT$=C1TITLE2$
13080  ERASE MATRIXOUT$
13090  DIM MATRIXOUT$ (NAGESOUT,NYEARSOUT)
13100    FOR I=1 TO NAGESOUT
13110      FOR J=1 TO NYEARSOUT
13120        MATRIXOUT$(I,J)=C1NATRIX$(I,J)
13130    NEXT J,I
13140 RETURN
13200 REM ***Numbers at age, midyear***
13210  CLS
13220  PRINT "Estimates for population numbers at age at midyear"
13230  PRINT "have been calculated."
13240  PRINT "Press any key to display output menu."
13250  A$=INKEY$ : IF A$="" GOTO 13250
13260  TITLE1OUT$=C2TITLE1$
13270  TITLE2OUT$=C2TITLE2$
13280  ERASE MATRIXOUT$
13290  DIM MATRIXOUT$ (NAGESOUT,NYEARSOUT)
13300    FOR I=1 TO NAGESOUT
13310      FOR J=1 TO NYEARSOUT
13320        MATRIXOUT$(I,J)=C2NATRIX$(I,J)
13330    NEXT J,I
13340 RETURN
13400 REM ***biomass, midyear or fishable***
13402  CLS
13404    IF OUTFLAGC3<>1 THEN PRINT CHR$(7) "Not defined. Returning to calculation menu..." : FOR I=1 TO 3000 : NEXT : RETURN 10600
13406  PRINT "Midyear or Fishable biomass? (M/F)   ";
13408  MF$=INKEY$ : IF MF$="" GOTO 13408
13409  PRINT MF$
13410    IF (MF$="M") OR (MF$="m") GOTO 13428
13412    IF (MF$="F") OR (MF$="f") GOTO 13414 ELSE GOTO 13406
13414    IF OUTFLAGIB<>1 THEN PRINT CHR$(7) "PR vector not defined. Returning to calculation menu...." : FOR I=1 TO 3000 : NEXT :
         RETURN 10600
13416    FOR I=1 TO NAGESOUT : FOR J=1 TO NYEARSOUT
13418      C3NATRIX$(I,J)=C2NATRIX$(I,J)+I2NATRIX$(I,J)+IBNATRIX$(I,I)
13420    NEXT J,I
13422  C3TITLE1$="Fishable biomass"
13424  PRINT : PRINT "Estimates for fishable biomass have been calculated."
13426 GOTO 13446
13428    IF (LEFT$(C3TITLE1$,1)="M") GOTO 13438
13430    FOR I=1 TO NAGESOUT : FOR J=1 TO NYEARSOUT
13432      C3NATRIX$(I,J)=C2NATRIX$(I,J)+I2NATRIX$(I,J)
13434    NEXT J,I
13436  C3TITLE1$="Midyear biomass"
13438  PRINT "Estimates for biomass at midyear have been calculated."
13440  PRINT "Press any key to display output menu."
13450  A$=INKEY$ : IF A$="" GOTO 13450
13460  TITLE1OUT$=C3TITLE1$
13470  TITLE2OUT$=C3TITLE2$
13480  ERASE MATRIXOUT$
13490  DIM MATRIXOUT$ (NAGESOUT,NYEARSOUT)
13500    FOR I=1 TO NAGESOUT
13510      FOR J=1 TO NYEARSOUT
13520        MATRIXOUT$(I,J)=C3NATRIX$(I,J)
13530    NEXT J,I
13540 RETURN
13600 REM ***Age-specific fishing mortalities***
13610  CLS
13620  PRINT "Estimates for age specific fishing mortalities "
13630  PRINT "have been calculated."
13640  PRINT "Press any key to display output menu."
13650  A$=INKEY$ : IF A$="" GOTO 13650
13660  TITLE1OUT$=C4TITLE1$
13670  TITLE2OUT$=C4TITLE2$
```

```
13680  ERASE MATRIXOUT$
13690  DIM MATRIXOUT$ (NAGESOUT,NYEARSOUT)
13700    FOR I=1 TO NAGESOUT
13710      FOR J=1 TO NYEARSOUT
13720        MATRIXOUT$(I,J)=C#MATRIX$(I,J)
13730    NEXT J,I
13740  RETURN
14000  REM ***Total numbers, beginning of year***
14010  CLS
14020  PRINT "Specify interval of ages to be summed for beginning of year "
14030  PRINT "population numbers? (Y/N)   ";
14040  A$=INKEY$ : IF A$="" GOTO 14040
14045  PRINT A$
14050    IF (A$="Y") OR (A$="y") GOTO 14070
14060    IF (A$="N") OR (A$="n") THEN FA=FAGEOUT : LA=FAGEOUT+NAGESOUT-1 : GOTO 14080 ELSE GOTO 14020
14070  PRINT : INPUT "Enter first age to be summed:   ",FA : INPUT "Enter last age to be summed:   ",LA
14080  FAIND=FA-FAGEOUT+1 : LAIND=LA-FAGEOUT+1
14090  IF EFLAGC5=1 THEN ERASE C5MATRIX$
14100  EFLAGC5=1
14110  DIM C5MATRIX$ (1,NYEARSOUT)
14120    FOR J=1 TO NYEARSOUT
14125      SUM$=0
14140        FOR I=FAIND TO LAIND
14150          SUM$=SUM$+C1MATRIX$(I,J)
14160        NEXT I
14170      C5MATRIX$(1,J)=SUM$
14180    NEXT J
14190  C5TITLE1$="Total N, beg. year"
14200  C5TITLE2$=GENTITLE$+" - AGES "+STR$(FA)+" TO "+STR$(LA)
14210  PRINT "Press any key to display output menu."
14220  A$=INKEY$ : IF A$="" GOTO 14220
14230  VYFLAG=1 : OUTFLAGC5=1
14240  TITLE1OUT$=C5TITLE1$
14250  TITLE2OUT$=C5TITLE2$
14260  ERASE MATRIXOUT$
14270  DIM MATRIXOUT$ (1,NYEARSOUT)
14280    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=C5MATRIX$(1,J) : NEXT
14290  RETURN
15000  REM ***Total numbers, midyear***
15010  CLS
15020  PRINT "Specify interval of ages to be summed for midyear "
15030  PRINT "population numbers? (Y/N)   ";
15040  A$=INKEY$ : IF A$="" GOTO 15040
15050  PRINT A$
15060    IF (A$="Y") OR (A$="y") GOTO 15080
15070    IF (A$="N") OR (A$="n") THEN FA=FAGEOUT : LA=FAGEOUT+NAGESOUT-1 : GOTO 15090 ELSE GOTO 15020
15080  PRINT : INPUT "Enter first age to be summed:   ",FA : INPUT "Enter last age to be summed:   ",LA
15090  FAIND=FA-FAGEOUT+1 : LAIND=LA-FAGEOUT+1
15100  IF EFLAGC6=1 THEN ERASE C6MATRIX$
15110  EFLAGC6=1
15120  DIM C6MATRIX$ (1,NYEARSOUT)
15130    FOR J=1 TO NYEARSOUT
15140      SUM$=0
15150        FOR I=FAIND TO LAIND
15160          SUM$=SUM$+C2MATRIX$(I,J)
15170        NEXT I
15180      C6MATRIX$(1,J)=SUM$
15190    NEXT J
15200  C6TITLE1$="Total N, midyear"
15210  C6TITLE2$=GENTITLE$+" - AGES "+STR$(FA)+" TO "+STR$(LA)
15220  PRINT "Press any key to display output menu."
15230  A$=INKEY$ : IF A$="" GOTO 15230
15240  VYFLAG=1 : OUTFLAGC6=1
15250  TITLE1OUT$=C6TITLE1$
15260  TITLE2OUT$=C6TITLE2$
15270  ERASE MATRIXOUT$
```

```
15280  DIM MATRIXOUT$ (1,NYEARSOUT)
15290    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=C4MATRIX$(1,J) : NEXT
15300 RETURN
16000 REM ***Total biomass, midyear or fishable***
16010  CLS
16020    IF OUTFLAGC3(>)1 THEN PRINT CHR$(7) "Not defined. Returning to calculation menu...." : FOR I=1 TO 3000 : NEXT : GOTO 10600
16025    IF (MF$="F") OR (MF$="f") THEN FA=FAGEOUT : LA=FAGEOUT+NAGESOUT-1 : PRINT "Calculating total fishable biomass..." :
         GOTO 16100
16030  PRINT "Specify interval of ages to be summed for midyear "
16040  PRINT "population biomass? (Y/N)   ";
16050  A$=INKEY$ : IF A$="" GOTO 16050
16060  PRINT A$
16070    IF (A$="Y") OR (A$="y") GOTO 16090
16080    IF (A$="N") OR (A$="n") THEN FA=FAGEOUT : LA=FAGEOUT+NAGESOUT-1 : GOTO 16100 ELSE GOTO 16030
16090  PRINT : INPUT "Enter first age to be summed:   ",FA : INPUT "Enter last age to be summed:   ",LA
16100  FAIND=FA-FAGEOUT+1 : LAIND=LA-FAGEOUT+1
16110  IF EFLAGC7=1 THEN ERASE C7MATRIX$
16120  EFLAGC7=1
16130  DIM C7MATRIX$ (1,NYEARSOUT)
16140    FOR J=1 TO NYEARSOUT
16150      SUM$=0
16160        FOR I=FAIND TO LAIND
16170          SUM$=SUM$+C3MATRIX$(I,J)
16180        NEXT I
16190      C7MATRIX$(1,J)=SUM$
16200    NEXT J
16210    IF (MF$="M") OR (MF$="m") THEN C7TITLE1$="Total B, midyear" ELSE C7TITLE1$="Total B, fishable"
16220  C7TITLE2$=GENTITLE$+" - AGES "+STR$(FA)+" TO "+STR$(LA)
16230  PRINT "Press any key to display output menu."
16240  A$=INKEY$ : IF A$="" GOTO 16240
16250  VFFLAG=1 : OUTFLAGC7=1
16260  TITLE1OUT$=C7TITLE1$
16270  TITLE2OUT$=C7TITLE2$
16280  ERASE MATRIXOUT$
16290  DIM MATRIXOUT$ (1,NYEARSOUT)
16300    FOR J=1 TO NYEARSOUT : MATRIXOUT$(1,J)=C7MATRIX$(1,J) : NEXT
16310 RETURN
17000 REM ***Mean fishing mortalities***
17010  CLS
17020  PRINT "Weighted or unweighted mean fishing mortalities? (W/U)   ";
17030  WEIGHTING$=INKEY$ : IF WEIGHTING$="" GOTO 17030
17040  PRINT WEIGHTING$
17050    IF (WEIGHTING$="W") OR (WEIGHTING$="w") OR (WEIGHTING$="U") OR (WEIGHTING$="u") GOTO 17060 ELSE GOTO 17020
17060  PRINT "Specify interval of ages to be averaged for fishing mortalities? (Y/N)   ";
17070  A$=INKEY$ : IF A$="" GOTO 17070
17080  PRINT A$
17090    IF (A$="Y") OR (A$="y") GOTO 17110
17100    IF (A$="N") OR (A$="n") THEN FA=FAGEOUT : LA=FAGEOUT+NAGESOUT-1 : GOTO 17120 ELSE GOTO 17060
17110  PRINT : INPUT "Enter first age to be averaged:   ",FA : INPUT "Enter last age to be averaged:   ",LA
17120  FAIND=FA-FAGEOUT+1 : LAIND=LA-FAGEOUT+1
17130  IF EFLAGC8=1 THEN ERASE C8MATRIX$
17140  EFLAGC8=1
17150  DIM C8MATRIX$ (1,NYEARSOUT)
17160    IF (WEIGHTING$="U") OR (WEIGHTING$="u") GOTO 17170 ELSE GOTO 17230
17170    FOR J=1 TO NYEARSOUT
17180      SUM$=0
17190        FOR I=FAIND TO LAIND
17200          SUM$=SUM$+C4MATRIX$(I,J)
17210        NEXT I
17220      C8MATRIX$(1,J)=SUM$/(LAIND-FAIND+1)
17230    NEXT J
17240 GOTO 17330
17250    FOR J=1 TO NYEARSOUT
17260      SUM1$=0 : SUM2$=0
17270        FOR I=FAIND TO LAIND
17280          SUM1$=SUM1$+(C1MATRIX$(I,J)*C4MATRIX$(I,J))
```

```
17290          SUM2#=SUM2#+C1#MATRIX#(I,J)
17300        NEXT I
17310       C0MATRIX#(I,J)=SUM1#/SUM2#
17320     NEXT J
17330     IF (WEIGHTING#="U") OR (WEIGHTING#="u") THEN C0TITLE1#="Unweighted mean F" ELSE C0TITLE1#="Weighted mean F"
17340  C0TITLE2#=GENTITLE#+" - AGES "+STR#(FA)+" TO "+STR#(LA)
17350  PRINT "Press any key to display output menu."
17360  A#=INKEY# : IF A#="" GOTO 17360
17370  VYFLAG=1 : OUTFLAGC#=1
17380  TITLE1OUT#=C0TITLE1#
17390  TITLE2OUT#=C0TITLE2#
17400  ERASE MATRIXOUT#
17410  DIM MATRIXOUT# (1,NYEARSOUT)
17420    FOR J=1 TO NYEARSOUT : MATRIXOUT#(1,J)=C0MATRIX#(1,J) : NEXT
17430 RETURN
18000 REM ***Overall fishing mortalities***
18010  CLS
18020  PRINT "Specify interval of ages to be included in calculation"
18030  PRINT "of overall fishing mortality? (Y/N)   ";
18040  A#=INKEY# : IF A#="" GOTO 18040
18050  PRINT A#
18060    IF (A#="Y") OR (A#="y") GOTO 18080
18070    IF (A#="N") OR (A#="n") THEN FA=FAGEOUT : LA=FAGEOUT+NAGESOUT-1 : GOTO 18090 ELSE GOTO 18020
18080  PRINT : INPUT "Enter first age to be included in calculations:   ",FA : INPUT
       "Enter last age to be included in calculations:   ",LA
18090  FAIND=FA-FAGEOUT+1 : LAIND=LA-FAGEOUT+1
18100  IF EFLAGC9=1 THEN ERASE C9#MATRIX#
18110  EFLAGC9=1
18120  DIM C9#MATRIX# (1,NYEARSOUT)
18130    FOR J=1 TO (NYEARSOUT-1)
18140      SUM1#=0 : SUM2#=0
18150        FOR I=FAIND TO (LAIND-1)
18160          SUM1#=SUM1#+C1#MATRIX#(I,J)
18170          SUM2#=SUM2#+C1#MATRIX#(I+1,J+1)
18180        NEXT I
18190      C9#MATRIX#(1,J)=(LOG(SUM1#/SUM2#))-#MORT
18200    NEXT J
18210  C9TITLE1#="Overall F"
18220  C9TITLE2#=GENTITLE#+" - AGES "+STR#(FA)+" TO "+STR#(LA)
18230  PRINT : PRINT "NOTE: The value for" FYEAROUT+NYEARSOUT-1 "is not defined. It has been set to zero." : PRINT
18240  PRINT "Press any key to display output menu."
18250  A#=INKEY# : IF A#="" GOTO 18250
18260  VYFLAG=1 : OUTFLAGC#=1
18270  TITLE1OUT#=C9TITLE1#
18280  TITLE2OUT#=C9TITLE2#
18290  ERASE MATRIXOUT#
18300  DIM MATRIXOUT# (1,NYEARSOUT)
18310    FOR J=1 TO NYEARSOUT : MATRIXOUT#(1,J)=C9#MATRIX#(1,J) : NEXT
18320 RETURN
19000 REM ***Partial recruitment matrices***
19010  CLS : PRINT "My silicon synapses are firing (calculating raw PR matrix)..."
19020  IF EFLAGC10=1 THEN ERASE C10#MATRIX#
19030  EFLAGC10=1
19040  DIM C10#AIVEC# (1,NYEARSOUT)
19050  DIM C10#MATRIX# (NAGESOUT,NYEARSOUT)
19060    FOR J=1 TO NYEARSOUT
19070      C10#AIVEC#(1,J)=-100
19080        FOR I=1 TO NAGESOUT
19090          IF C#MATRIX#(I,J)>C10#AIVEC#(1,J) THEN C10#AIVEC#(1,J)=C#MATRIX#(I,J)
19100    NEXT I,J
19110    FOR J=1 TO NYEARSOUT
19120      FOR I=1 TO NAGESOUT
19130        C10#MATRIX#(I,J)=C#MATRIX#(I,J)/C10#AIVEC#(1,J)
19140    NEXT I,J
19150  ERASE C10#AIVEC# : OUTFLAGC10=1 : CLS
19160  PRINT "The following options are available for manipulation of partial recruitments:" : PRINT
```

```
19170  PRINT "F1. Calculate standardized arithmetic mean PR vector for a specified"
19180  PRINT "    interval of years."
19190  PRINT "F2. Divide PR matrix by value used to standardize vector calculated in F1."
19200  PRINT "F3. Calculate standardized geometric mean PR vector for a specified"
19210  PRINT "    interval of years."
19220  PRINT "F4. Divide PR matrix by value used to standardize vector calculated in F3."
19225  PRINT "F5. Output raw PR matrix." : PRINT
19230  PRINT "NOTE": Only the most recent option chosen will remain in memory for output."
19240  C$=INKEY$ : IF C$="" THEN 19240
19250  CHECK1=ASC(C$) : LOCATE 24,1,0 : PRINT SPACE$(79);
19260    IF CHECK1<>0 THEN LOCATE 24,1 : PRINT CHR$(7) "Please press a valid F-key."; : GOTO 19240
19270  EXTCODE=ASC(MID$(C$,2))-58
19275    IF EXTCODE=5 THEN C1OTITLE1$="Raw PR matrix" : GOTO 19570
19280    IF EXTCODE>5 THEN LOCATE 24,1 : PRINT CHR$(7) "Please press a valid F-key."; : GOTO 19240
19290  REM ***Mean PR***
19300  CLS
19310  INPUT "Enter first year to be averaged:   ",FY
19320  INPUT "Enter last year to be averaged:    ",LY
19330  FYIND=FY-FYEAROUT+1 : LYIND=LY-FYEAROUT+1
19340  DIM C1OMAIVEC$(NAGESOUT,1)
19350  NYRAVG=LYIND-FYIND+1
19360    ON EXTCODE GOTO 19370,19370,19430,19430
19365  REM ***Arithmetic mean***
19370    FOR I=1 TO NAGESOUT
19380      SUM$=0
19390        FOR J=FYIND TO LYIND : SUM$=SUM$+C1ONATRI3$(I,J) : NEXT J
19400      C1OMAIVEC$(I,1)=SUM$/NYRAV$
19410    NEXT I
19420  GOTO 19480
19425  REM ***Geometric mean***
19430    FOR I=1 TO NAGESOUT
19440      SUM$=0
19450        FOR J=FYIND TO LYIND : SUM$=SUM$+LOG(C1ONATRI3$(I,J)) : NEXT J
19460      C1OMAIVEC$(I,1)=EXP(SUM$/NYRAV$)
19470    NEXT I
19480  MAI$=-100
19490    FOR I=1 TO NAGESOUT : IF C1OMAIVEC$(I,1)>MAI$ THEN MAI$=C1OMAIVEC$(I,1
19500    NEXT
19510    ON EXTCODE GOTO 19670,19515,19670,19515
19515  ERASE C1OMAIVEC$
19520    FOR J=1 TO NYEARSOUT                              '***Output matrix***
19530      FOR I=1 TO NAGESOUT
19540        C1ONATRI3$(I,J)=C1ONATRI3$(I,J)/MAI$
19550      NEXT I,J
19560    IF EXTCODE=2 THEN C1OTITLE1$="PR matrix (from AR)" ELSE C1OTITLE1$="PR matrix (from BR)"
19570  C1OTITLE2$=IDENTITLE$
19580  TITLE1OUT$=C1OTITLE1$
19590  TITLE2OUT$=C1OTITLE2$
19600  ERASE MATRIXOUT$
19610  DIM MATRIXOUT$ (NAGESOUT,NYEARSOUT)
19620    FOR I=1 TO NAGESOUT
19630      FOR J=1 TO NYEARSOUT
19640        MATRIXOUT$(I,J)=C1ONATRI3$(I,J)
19650    NEXT J,I
19660  RETURN
19670  ERASE C1ONATRI3$ : DIM C1ONATRI3$(NAGESOUT,1)    '***Output vector***
19680  IF EFLAG3$=1 THEN ERASE IBMATRI3$
19690  EFLAG3$=1 : OUTFLAG3$=1 : PRFLAG=1 : VPY=1 : DIM IBMATRI3$(NAGESOUT,1)
19700    FOR I=1 TO NAGESOUT : C1ONATRI3$(I,1)=C1OMAIVEC$(I,1)/MAI$ : IBMATRI3$(I,1)=C1ONATRI3$(I,1) : NEXT
19705  ERASE C1OMAIVEC$
19710  PRINT : PRINT "Standardized partial recruitment vector has been calculated."
19720  PRINT "This vector may be used in input menu item 3 to generate a new F for last year."
19725  PRINT "NOTE: The vector just calculated has replaced the previous input vector."
19730    IF EXTCODE=1 THEN C1OTITLE1$="Arith. mean PR" ELSE C1OTITLE1$="Geom. mean PR"
19740  C1OTITLE2$=IDENTITLE$+" - YEARS "+STR$(FY)+" TO "+STR$(LY)
19745  IBTITLE1$=C1OTITLE1$ : IBTITLE2$=C1OTITLE2$
```

```
19750  PRINT : PRINT "Press any key to display output menu."
19760  A$=INKEY$ : IF A$="" GOTO 19760
19770  VAFLAG=1 : GOSUB 9730
19780  TITLE1OUT$=C10TITLE1$
19790  TITLE2OUT$=C10TITLE2$
19800  ERASE MATRIXOUT$
19810  DIM MATRIXOUT$ (MAGESOUT,1)
19820    FOR I=1 TO MAGESOUT
19830      MATRIXOUT$(I,1)=C10MATRIX$(I,1)
19840    NEXT I
19850 RETURN
20000 REM ***OUTPUT***
20010  CLS
20020  PRINT "        OUTPUT MENU FOR COHORT ANALYSIS "; : COLOR 7,0 : PRINT "(" NOWOUT$ ")" : COLOR 2,0
20030  PRINT
20040  PRINT "F1. Print on screen"
20050  PRINT "F2. Print on line printer (will print filename if F3 was last item chosen"
20055  PRINT "    and output menu was invoked from a calculation menu item)"
20060  PRINT "F3. Output on diskette"
20065    IF NOWOUT$="GENERAL" GOTO 20093
20070  PRINT "F4. Go to PLOT & FIT program (N.B. Only valid when coming from"
20080  PRINT "    a calculation menu item. The last item executed will"
20090  PRINT "    be passed to PLOT & FIT, as well as the independent index."
20092  PRINT "    Arrays defined in the calculation menu are then erased.)"
20093  PRINT "F5. Go to data ENTRY program"
20095  PRINT "F6. END program"
20100  PRINT "CTRL F1. INPUT MENU"
20110  PRINT "CTRL F2. CALCULATION MENU"
20120 RETURN
20200 REM ***from menu***
20205  COLOR 2,0 : NOWOUT$="GENERAL"
20210 GOSUB 20000
20220  C$=INKEY$ : IF C$="" THEN 20220
20230  CHECK1=ASC(C$)
20240  LOCATE 24,1,0 : PRINT SPACE$(79);
20250    IF CHECK1<>0 THEN LOCATE 24,1,0 : PRINT CHR$(7) "Please press a valid F-key."; : GOTO 20220
20260  EXTCODE=ASC(MID$(C$,2))
20270    ON EXTCODE-58 GOSUB 22000,24000,26000
20280    IF (EXTCODE-58)<4 THEN 20200
20290    IF EXTCODE=62  THEN CLS : PRINT "Chaining to ENTRY program..." : CLEAR : CHAIN "ENTRY"
20300    IF EXTCODE=64  GOTO  29000
20310    IF EXTCODE=94  THEN VAFLAG=0 : VYFLAG=0 : CHAIN "cohort",100,ALL
20320    IF (EXTCODE<95) AND (CALCFLAG=1) THEN VAFLAG=0 : VYFLAG=0 : GOTO 10000
20330    IF (EXTCODE=95) AND (CALCFLAG=0) THEN VAFLAG=0 : VYFLAG=0 : GOTO 10600
20340  LOCATE 24,1,0 : PRINT CHR$(7) "Please press a valid F-key."; : GOTO 20220
20400 REM ***from menu item***
20402    IF VAFLAG=1 THEN GOSUB 9700
20404    IF VYFLAG=1 THEN GOSUB 9750
20405  COLOR 2,0 : NOWOUT$=TITLE1OUT$
20410 GOSUB 20000
20420  C$=INKEY$ : IF C$="" THEN 20420
20430  CHECK1=ASC(C$)
20440  LOCATE 24,1,0 : PRINT SPACE$(79);
20450    IF CHECK1<>0 THEN LOCATE 24,1,0 : PRINT CHR$(7) "Please press a valid F-key."; : GOTO 20420
20460  EXTCODE=ASC(MID$(C$,2))
20470    IF EXTCODE=59  THEN CLS : GOSUB 23000 : GOTO 20405
20480    IF EXTCODE=60  THEN CLS : GOSUB 25000 : GOTO 20405
20490    IF EXTCODE=61  THEN CLS : GOSUB 27000 : GOTO 20405
20500    IF EXTCODE=62  GOTO 20000
20505    IF EXTCODE=63  THEN CLS : PRINT "Chaining to ENTRY program..." : CLEAR : CHAIN "ENTRY"
20510    IF EXTCODE=64  GOTO 29000
20520    IF EXTCODE=94  THEN GOSUB 20600 : CHAIN "cohort",100,ALL
20530    IF (EXTCODE<95) AND (CALCFLAG=1) THEN GOSUB 20600 : GOTO 10000
20532    IF (EXTCODE=95) AND (CALCFLAG=0) THEN GOSUB 20600 : GOTO 10600
20540  LOCATE 24,1,0 : PRINT CHR$(7) "Please press a valid F-key."; : GOTO 20420
```

```
20600 REM ***Reset dimensions***
20610   IF VAFLAG=1 THEN GOSUB 9730 : VAFLAG=0
20620   IF VYFLAG=1 THEN GOSUB 9780 : VYFLAG=0
20630 RETURN
21000 REM ***Items available for printing***
21010   IF OUTFLAG11 =1 THEN LOCATE 14,1 ,0 : OUTVEC%(1)=1 : PRINT "I1.  Catch at age matrix"
21020   IF OUTFLAG12 =1 THEN LOCATE 15,1 ,0 : OUTVEC%(2)=1 : PRINT "I2.  Weight at age matrix"
21030   IF OUTFLAG13 =1 THEN LOCATE 16,1 ,0 : OUTVEC%(3)=1 : PRINT "I3.  Fishing mortality, last year"
21040   IF OUTFLAG14 =1 THEN LOCATE 17,1 ,0 : OUTVEC%(4)=1 : PRINT "I4.  Fishing mortality, last age"
21050   IF OUTFLAG15 =1 THEN LOCATE 18,1 ,0 : OUTVEC%(5)=1 : PRINT "I5.  Natural mortality"
21060   IF OUTFLAG18 =1 THEN LOCATE 19,1 ,0 : OUTVEC%(6)=1 : PRINT "I6.  Partial recruitment (input)"
21070   IF OUTFLAG19 =1 THEN LOCATE 20,1 ,0 : OUTVEC%(7)=1 : PRINT "I7.  Independent index"
21075 LOCATE 24,1,0 : PRINT "IA.  Print all input items";
21080   IF OUTFLAGC1 =1 THEN LOCATE 14,40,0 : OUTVEC%(8)=1 : PRINT "C1.  Numbers at age, beginning of year"
21090   IF OUTFLAGC2 =1 THEN LOCATE 15,40,0 : OUTVEC%(9)=1 : PRINT "C2.  Numbers at age, midyear"
21100   IF OUTFLAGC3 =1 THEN LOCATE 16,40,0 : OUTVEC%(10)=1 : PRINT "C3.  Biomass, midyear or fishable"
21110   IF OUTFLAGC4 =1 THEN LOCATE 17,40,0 : OUTVEC%(11)=1 : PRINT "C4.  Age specific fishing mortalities"
21120   IF OUTFLAGC5 =1 THEN LOCATE 18,40,0 : OUTVEC%(12)=1 : PRINT "C5.  Total numbers, beginning of year"
21130   IF OUTFLAGC6 =1 THEN LOCATE 19,40,0 : OUTVEC%(13)=1 : PRINT "C6.  Total numbers, midyear"
21140   IF OUTFLAGC7 =1 THEN LOCATE 20,40,0 : OUTVEC%(14)=1 : PRINT "C7.  Total biomass, midyear or fishable"
21150   IF OUTFLAGC8 =1 THEN LOCATE 21,40,0 : OUTVEC%(15)=1 : PRINT "C8.  Mean fishing mortalities"
21160   IF OUTFLAGC9 =1 THEN LOCATE 22,40,0 : OUTVEC%(16)=1 : PRINT "C9.  Overall fishing mortalities"
21170   IF OUTFLAGC10=1 THEN LOCATE 23,40,0 : OUTVEC%(17)=1 : PRINT "C10. Partial recruitment (calc'd)"
21175 LOCATE 24,40,0 : PRINT "CA.  Print all calculation items";
21180   IF EXITCODE=61 THEN LOCATE 10,1 : PRINT SPACE$(30);
21190   IF (EXITCODE=59) OR (EXITCODE=61) THEN LOCATE 24,1,0 : PRINT SPACE$(75);
21200 LOCATE 11,1 : INPUT "Enter code for item to be output:   ",ITEMCODE$
21210   IF (ITEMCODE$="IA") OR (ITEMCODE$="ia") THEN CLS : ALLOUTFLAG=1 : GOSUB 21800 : RETURN
21215   IF (ITEMCODE$="CA") OR (ITEMCODE$="ca") THEN CLS : ALLOUTFLAG=1 : GOSUB       21900 : RETURN
21220 WHATOUT%=VAL(MID$(ITEMCODE$,2))
21225   IF (LEFT$(ITEMCODE$,1)="I") OR (LEFT$(ITEMCODE$,1)="i") THEN GOSUB 21240 : RETURN
21230   IF (LEFT$(ITEMCODE$,1)="C") OR (LEFT$(ITEMCODE$,1)="c") THEN GOSUB 21270 : RETURN
21240     ON WHATOUT% GOSUB 21260,21260, 9700, 9750,21260, 9700,21300
21245     ON WHATOUT% GOSUB  1360, 2095, 3930, 4750,21260, 7170, 8170
21250     ON WHATOUT% GOSUB 21500,21500,21500,21500,21700,21500,21500
21255     ON WHATOUT% GOSUB 21260,21260, 9730, 9780,21260, 9730,21310
21260   RETURN
21270     ON WHATOUT% GOSUB 21290,21290,21290,21290, 9750, 9750, 9750, 9750, 9750,21320
21275     ON WHATOUT% GOSUB 13060,13260,13460,13660,14240,15250,16260,17380,18270,21330
21280     GOSUB 21500
21285     ON WHATOUT% GOSUB 21290,21290,21290,21290, 9780, 9780, 9780, 9780, 9780,21340
21290   RETURN
21300     IF VAFLAG=1 THEN GOSUB 9750
21305   RETURN
21310     IF VAFLAG=1 THEN GOSUB 9780
21315   RETURN
21320     IF PRFLAG=1 THEN GOSUB 9700
21325   RETURN
21330     IF PRFLAG=1 THEN GOSUB 19780 ELSE GOSUB 19580
21335   RETURN
21340     IF PRFLAG=1 THEN GOSUB 9730
21345   RETURN
21380 BEEP : LOCATE 11,1 : PRINT SPACE$(79); : GOTO 21200
21500 REM ***Screen, printer, or diskette output***
21510   ON EXITCODE-58 GOSUB 22900,24900,27000 : RETURN
21700 REM ***Output natural mortality***
21710   IF EXITCODE=59 THEN GOSUB 22300 : RETURN
21720   IF EXITCODE=60 THEN GOSUB 24300 : RETURN
21730   IF EXITCODE=61 THEN RETURN 21380
21800 REM ***Print all available input menu items***
21810   FOR NOWPRINT=1 TO 7
21820     IF OUTVEC%(NOWPRINT)=1 THEN WHATOUT%=NOWPRINT : GOSUB 21240
21830   NEXT
21840 ALLOUTFLAG=0 : BS$="" : RETURN
21900 REM ***Print all available calculation menu items***
21910   FOR NOWPRINT=8 TO 17
21920     IF OUTVEC%(NOWPRINT)=1 THEN WHATOUT%=NOWPRINT-7 : GOSUB 21270
```

```
21930    NEXT
21940  ALLOUTFLAG=0 : BS$="" : RETURN
22000 REM ***Print on screen***
22010  CLS : VAFLAG=0 : VVFLAG=0 : DISKFLAG=0
22020  PRINT "The following is available for screen display. The numbering is almost"
22030  PRINT "the same as in the input menu (I) and calculation menu (C). To select"
22040  PRINT "an item for printing, enter the code preceding the item (eg. C6).
22050  PRINT "Matrices with <=8 years of data will be printed on one screen, "
22060  PRINT "but matrices with >8 years will need at least two screens. To "
22070  PRINT "print the next screen, press 'Pg Up'. To reprint the previous"
22080  PRINT "screen, press 'Pg Dn'. Press 'X' to exit to output menu. Each item"
22090  PRINT "is printed with a title, ages and years (if applicable)."
22100 GOSUB 21000 : RETURN
22300 REM ***Print natural mortality***
22310  CLS
22320  PRINT "Natural mortality is :   "!NMORT
22330  PRINT
22340  PRINT "Press any key to return to output menu."
22350  C$=INKEY$ : IF C$="" THEN 22350
22360 RETURN
22900  LOCATE 12,1
23000 REM ***Display data***
23010  LOCATE 12,1 : PRINT SPACE$(79)! : LOCATE 12,1
23020  PRINT "Enter desired number of decimal places for " TITLE1OUT$ " (0-4):   "! : DISKFLAG=0
23030  DECIOUT$=INKEY$ : IF DECIOUT$="" THEN 23030
23040  PRINT DECIOUT$ : DECIOUT=VAL(DECIOUT$)
23050    IF DECIOUT=0 THEN SOFTPR$="########" : GOTO 23070
23060    IF (DECIOUT>0) AND (DECIOUT<=4) THEN SOFTPR$=STRING$(8-DECIOUT,"#")+"."+STRING$(DECIOUT,"#") ELSE GOTO 23010
23070  PAGE=1 : OUTFLAG=0
23080  PAGES=CINT((NYEARSOUT/8)+.45)
23090    ON PAGE GOSUB 23760,23800,23840 : GOSUB 23130 : GOTO 23120
23120    IF OUTFLAG=1 THEN RETURN ELSE GOTO 23090
23130 REM ***Print matrix***
23140  LYR=LY+FYEAROUT-1
23150  HYR=HY+FYEAROUT-1
23160  COLOR 2,0
23170  CLS
23180  COLOR 1,7
23190  PRINT SPACE$(79);
23200  LOCATE 1,1,0 : PRINT TITLE1OUT$; : IF (TITLE2OUT$=" ") OR (TITLE2OUT$="") THEN 23210 ELSE PRINT " : " LEFT$(TITLE2OUT$,70)!
       " (RUN #";STR$(RUNNUM)!")"
23210  COLOR 7,1
23220  LOCATE 2,1 : PRINT "  YEAR "
23230    FOR I=LY TO HY
23240      DUMMY=I
23250        IF LY=9 THEN DUMMY=DUMMY-8
23260        IF LY=17 THEN DUMMY=DUMMY-16
23270      PY=DUMMY+9-1
23280      PYR=FYEAROUT+I-1
23290      LOCATE 2,PY : PRINT USING "########";PYR
23300    NEXT
23310  PRINT
23320    IF NAGESOUT=1 THEN PRINT "       " ELSE PRINT "  AGE  "
23330    FOR J=1 TO NAGESOUT
23340      INAGE=FAGEOUT+J-1
23350        IF NAGESOUT=1 THEN PRINT "    "; ELSE PRINT USING "####";INAGE;
23360        IF J=NAGESOUT THEN PRINT "   "! ELSE PRINT "   "
23370    NEXT J
23380    FOR K=1 TO NAGESOUT
23390      ROWE=K+4
23400      LOCATE ROWE,8
23410        FOR L=LY TO HY
23420          EVENODD=(INT(L/2))*2
23430            IF L=EVENODD THEN COLOR 0,2 ELSE COLOR 2,0
23440          PRINT USING SOFTPR$;MATRIIOUT#(K,L)!
23450    NEXT L,K
```

```
23460 REM ***Change screen***
23470  PRINTING=-1
23480    IF (PAGE<PAGES) AND (LY=9) THEN HY=8
23490    IF (PAGE=PAGES) AND (LY=9) THEN HY=NYEARS-8
23500    IF (PAGE=PAGES) AND (LY=17) THEN HY=NYEARS-16
23510    WHILE PRINTING
23520      ED$=INKEY$ : IF ED$="" THEN 23520
23530        IF (ED$="X") OR (ED$="x") THEN GOSUB 23390 : GOTO 23570
23540      EDCODE=ASC(ED$)
23550        IF EDCODE=0 THEN GOSUB 23610 : GOTO 23570       'ED$=extended code
23560      BEEP
23570    WEND
23580 RETURN
23590 REM ***Finish Printing***
23600  PRINTING=0 : OUTFLAG=1 : RETURN
23610 REM ***Extended code***
23620  EXCODE=ASC(MID$(ED$,2))
23630  IF EXCODE=73  THEN GOSUB 23640 : RETURN      'Pg Up
23640  IF EXCODE=81  THEN GOSUB 23690 : RETURN      'Pg Dn
23650  BEEP : RETURN
23660 REM ***Page up***
23670  IF PAGE=PAGES THEN BEEP : RETURN
23680  PAGE=PAGE+1 : PRINTING=0 : RETURN
23690 REM ***Page down***
23700  IF PAGE=1 THEN BEEP : RETURN
23710  PAGE=PAGE-1 : PRINTING=0 : RETURN
23760 REM ***Page 1***
23770  LY=1
23780  IF PAGE<PAGES THEN HY=8 ELSE HY=NYEARSOUT
23790  RETURN
23800 REM ***Page 2***
23810  LY=9
23820  IF PAGE<PAGES THEN HY=16 ELSE HY=NYEARSOUT
23830  RETURN
23840 REM ***Page 3***
23850  LY=17
23860  HY=NYEARSOUT
23870  RETURN
24000 REM ***Hardcopy***
24010  CLS : VAFLAG=0 : VYFLAG=0
24020  PRINT "The following is available for hardcopy. The numbering is almost the"
24030  PRINT "same as in the input menu (I) and calculation menu (C). To select
24040  PRINT "an item for printing, enter the code preceding the item (eg. C6).
24050  PRINT "Each item is output with a title, ages and years (if applicable).
24060  PRINT "When using IA or CA option, default number of decimal places is"
24070  PRINT "zero for any population numbers or biomass options; three for all
24080  PRINT "other output."
24100 GOSUB 21000 : RETURN
24300 REM ***Print natural mortality***
24310  LPRINT : LPRINT : LPRINT
24320  LPRINT "Natural mortality is :   "INMORT
24330 RETURN
24900  LOCATE 12,1
25000 REM ***Hardcopy sub-routine***
25020  LPRINT
25030  LPRINT
25040    IF DISKFLAG=1 THEN LPRINT CHR$(18) "Name of diskette file is:   "; FILESTORE$ : DISKFLAG=0 : FILESTORE$="" : LPRINT
25041    IF ALLOUTFLAG=1 GOTO 25042 ELSE GOTO 25050
25042    IF DS$="" GOTO 25043 ELSE GOTO 25046
25043  PRINT "Default number of decimal places or Specify? (D/S):   ";
25044  DS$=INKEY$ : IF DS$="" THEN 25044
25045  PRINT DS$
25046    IF (DS$="D") OR (DS$="d") THEN DECIOUT=DEFDECI(NOWPRINT) : GOTO 25090
25047    IF (DS$="S") OR (DS$="s") THEN 25049 ELSE LOCATE 12,1 : PRINT SPACE$(79); : LOCATE 12,1 : GOTO 25043
25049  LOCATE 12,1 : PRINT SPACE$(79); : LOCATE 12,1
25050  PRINT "Enter desired number of decimal places for " TITLE1OUT$ " (0-4):   ";
```

```
25060 DECIOUT$=INKEY$ : IF DECIOUT$="" THEN 25060
25070 PRINT DECIOUT$ : DECIOUT=VAL(DECIOUT$)
25080   IF (DECIOUT<0) OR (DECIOUT>4) GOTO 25049
25090 MAXTESTOUT$=-99999999999
25100   FOR I=1 TO NAGESOUT
25110     FOR J=1 TO NYEARSOUT
25120       IF ABS(MATRIXOUT$(I,J))>MAXTESTOUT$ THEN MAXTESTOUT$=ABS(MATRIXOUT$(I,J))
25130   NEXT J,I
25140 MAXOUT=LEN(STR$(INT(ABS(MAXTESTOUT$))))-1
25150   IF DECIOUT=0 THEN MAXLENOUT=MAXOUT+2 : GOTO 25170
25160 MAXLENOUT=MAXOUT+DECIOUT+3
25170   IF MAXLENOUT<5 THEN MAXLENOUT=5
25180 NFIELDSOUT=INT(126/MAXLENOUT)
25190   IF DECIOUT=0 THEN HARDPROUT$=STRING$(MAXLENOUT,"0") : GOTO 25210
25200 HARDPROUT$=STRING$(MAXOUT+2,"0")+"."+STRING$(DECIOUT,"0")
25210 YEARPROUT$=STRING$(MAXLENOUT,"0")
25220   IF NYEARSOUT>NFIELDSOUT THEN BLOCKSOUT=2 ELSE BLOCKSOUT=1
25230 WIDTH "LPT1:",132
25240   IF BLOCKSOUT=1 THEN LYOUT=1 : HYOUT=NYEARSOUT : GOSUB 25260 : WIDTH "LPT1:",80 : RETURN
25250   IF BLOCKSOUT=2 THEN GOSUB 25500 : WIDTH "LPT1:",80 : RETURN
25260 REM ***Print data***
25270 LPRINT CHR$(15); TITLE1OUT$; : IF (TITLE2OUT$=" ") OR (TITLE2OUT$="") THEN LPRINT TAB(120) DATE$ ELSE LPRINT " : " TITLE2OUT$
      " (RUN #" STR$(RUNNUM) ")" TAB(120) DATE$
25280 LPRINT
25290 LPRINT
25300 LPRINT CHR$(15); "YEAR ";
25310   FOR I=LYOUT TO HYOUT
25320     YEAROUT=FYEAROUT+I-1
25330     LPRINT USING YEARPROUT$;YEAROUT;
25340   NEXT I
25350 LPRINT
25360 LPRINT
25370 IF NAGESOUT=1 THEN LPRINT "    " ELSE LPRINT "AGE  "
25380 LPRINT
25390   FOR K=1 TO NAGESOUT
25400     AGEOUT=FAGEOUT+K-1
25410       IF NAGESOUT=1 THEN LPRINT "  "; ELSE LPRINT USING "##";AGEOUT;
25420     LPRINT "   ";
25430       FOR L=LYOUT TO HYOUT
25440         LPRINT USING HARDPROUT$;MATRIXOUT$(K,L);
25450       NEXT L
25460     LPRINT
25470   NEXT K
25480 LPRINT CHR$(18)
25490 RETURN
25500 REM ***Print data in two or more blocks***
25510 LYOUT=1
25520 HYOUT=NFIELDSOUT
25530 GOSUB 25260
25540 LYOUT=LYOUT+NFIELDSOUT
25550 HYOUT=HYOUT+NFIELDSOUT
25560   WHILE HYOUT<NYEARSOUT
25570     GOSUB 25280
25580     LYOUT=LYOUT+NFIELDSOUT
25590     HYOUT=HYOUT+NFIELDSOUT
25600   WEND
25610 HYOUT=NYEARSOUT
25620 GOSUB 25280
25630 RETURN
26000 REM ***Output to disk***
26010 CLS : VAFLAG=0 : VYFLAG=0
26020 PRINT "The following is available for diskette output. The numbering is almost"
26030 PRINT "the same as in the input menu (I) and calculation menu (C). To select"
26040 PRINT "an item for output, enter the code preceding the item (eg. C6)."
26050 PRINT "Each item is output with a title, precision, number of ages,"
26060 PRINT "number of years, first age and first year."
26100 GOSUB 21000 : RETURN
```

```
27000 REM ***Disk output***
27010  CLS
27020  PRINT "Would you like single or double precision for diskette output? (S/D)   ";
27030  PRECISION$=INKEY$ : IF PRECISION$="" THEN 27030
27040  PRINT PRECISION$
27050    IF (PRECISION$="S") OR (PRECISION$="s") OR (PRECISION$="D") OR (PRECISION$="d") THEN 27060 ELSE GOTO 27020
27060    IF (PRECISION$="S") OR (PRECISION$="s") THEN GOSUB 27920
27070 GOSUB 27850
27080  INPUT "Enter filename:   ",FILENAME$
27090  PRINT "Insert data diskette into drive B."
27100  PRINT "Press any key to begin output to diskette."
27110  A$=INKEY$ : IF A$="" THEN 27110
27120  DOT=INSTR(FILENAME$,".")
27130    IF DOT=0 THEN FILESPEC$="B:"+FILENAME$+".DAT" ELSE FILESPEC$="B:"+FILENAME$
27140  ERRFLAG=1
27150 GOSUB 27350
27160    IF ERRFLAG=1 THEN GOSUB 27350
27170    OPEN FILESPEC$ FOR  OUTPUT AS #1
27180      PRINT#1, TITLE1OUT$
27190      PRINT#1, TITLE2OUT$
27200      PRINT#1, PRECISION$
27210      PRINT#1, NAGESOUT;NYEARSOUT;FAGEOUT;FYEAROUT;
27220        FOR I=1 TO NAGESOUT
27230          FOR J=1 TO NYEARSOUT
27240            IF (PRECISION$="D") OR (PRECISION$="d") THEN PRINT#1, MATRIXOUT#(I,J); ELSE TEMP!=MATRIXOUT#(I,J) : PRINT#1,TEMP!;
27250          NEXT J
27260        NEXT I
27270    CLOSE #1
27280    ON ERROR GOTO 0
27290  FILESTORE$=MID$(FILESPEC$,3)
27300  PRINT "Data has been stored on drive B as:   ";FILESTORE$
27310  PRINT "Press any key to return to menu."
27320  DISKFLAG=1
27330  A$=INKEY$ : IF A$="" THEN 27330
27340 RETURN
27350 REM ***Check if file exists***
27360    ON ERROR GOTO 27450
27370  ERRFLAG=0
27380  LN=CSRLIN
27390  LOCATE 23,1 : COLOR 0,0
27400  FILECHECK=0
27410  FILES FILESPEC$ : FILECHECK=1
27420    IF FILECHECK=1 GOTO 27480    'File already exists
27430  COLOR 2,0 : LOCATE LN+1,1
27440    ON ERROR GOTO 27770 : RETURN
27450    IF ERR=53 THEN RESUME 27430
27460  COLOR 2,0 : LOCATE LN+1,1
27470    ON ERROR GOTO 0
27480 ON ERROR GOTO 0 : COLOR 2,0 : LOCATE LN+1,1
27490  ERRFLAG=1
27500  CLS
27510  PRINT CHR$(7) "File already exists. Would you like to: "
27520  PRINT "  A. Enter a different filename?"
27530  PRINT "  B. Rename the existing file without changing the name you have just entered?"
27540  PRINT "  C. Write over the existing file?"
27550  PRINT
27560  PRINT "Press A, B, or C on the keyboard.   ";
27570  OPT$=INKEY$ : IF OPT$="" THEN 27570
27580  PRINT OPT$
27590    IF (OPT$="A") OR (OPT$="a") THEN GOSUB 27620 : GOTO 27760
27600    IF (OPT$="B") OR (OPT$="b") THEN GOSUB 27660 : GOTO 27760
27610    IF (OPT$="C") OR (OPT$="c") THEN GOSUB 27740 : GOTO 27760 ELSE GOTO 27560
27620 INPUT "Enter new filename:   ",FILENAME$     '***Option A***
27630  DOT=INSTR(FILENAME$,".")
27640    IF DOT=0 THEN FILESPEC$="B:"+FILENAME$+".DAT" ELSE FILESPEC$="B:"+FILENAME$
27650 RETURN
```

```
27660 INPUT "Enter new filename for existing file:   ",NEWFN$  '***Option B***
27670  DOT=INSTR(NEWFN$,".")
27680    IF DOT=0 THEN NEWFS$="B:"+NEWFN$+".DAT" ELSE NEWFS$="B:"+NEWFN$
27690    ON ERROR GOTO 27820
27700  NAME FILESPEC$ AS NEWFS$
27710  NEWFS$=MID$(NEWFS$,3)
27720  PRINT "New name for existing file is:   "NEWFS$
27730 RETURN
27740 KILL FILESPEC$           '***Option C***
27750 RETURN
27760    ON ERROR GOTO 27770 : RETURN
27770 REM ***Errors for full diskette***
27780  CLS
27790    IF (ERR=61) THEN PRINT CHR$(7) "Data diskette is full." : PRINT "Insert a new diskette into drive B." : KILL FILESPEC$ :
         RESUME 27810
27800    IF (ERR=67) THEN PRINT CHR$(7) "Data diskette has too many files." : PRINT "Insert a new diskette into drive B." :
         RESUME 27810
27810    ON ERROR GOTO 0 : GOTO 27100
27820 REM ***NAME statement error***
27830    IF ERR=58 THEN PRINT CHR$(7) "New filename already exists." : RESUME 27660
27840    ON ERROR GOTO 0
27850 REM ***Instructions***
27860  PRINT
27870  PRINT "Rules for naming of files are as described in the BASIC reference
27880  PRINT "manual. If no filename extension is included then the extension "
27890  PRINT "'.dat ' is added by default."
27900  PRINT
27910 RETURN
27920 REM ***Precision verification***
27930  PRINT CHR$(7)
27940  PRINT "WARNING: Single precision output will only store 7 significant digits."
27950  PRINT "Any numbers with more than 7 significant digits will be rounded off."
27960  PRINT "Press 'D ' if you would rather have double precision, if not,"
27970  PRINT "press RETURN." : PRINT
27980  PV$=INKEY$ : IF PV$="" THEN 27980
27990    IF (PV$="D") OR (PV$="d") THEN PRECISION$="D" : RETURN
27995    IF (PV$=CHR$(13)) THEN RETURN ELSE GOTO 27960
28000 REM ***Chain to PLOT & FIT***
28010  CLS : DISKFLAG=0
28020    IF OUTFLAG19=0 THEN PRINT CHR$(7) "Independent index is not defined. Returning to input menu...." : FOR I=1 TO 3000 : NEXT
         ELSE CHAIN"preplot",,ALL
28030    IF VAFLAG=1 THEN GOSUB 9730 : VAFLAG=0
28040    IF VYFLAG=1 THEN GOSUB 9780 : VYFLAG=0
28050 CHAIN "cohort",100,ALL
29000 REM ***Reset F-keys and end program***
29010  KEY 1,"list "
29020  KEY 2,"run"+CHR$(13)
29030  KEY 3,"load"+CHR$(34)
29040  KEY 4,"save"+CHR$(34)
29050  KEY 5,"cont"+CHR$(13)
29060  KEY 6,"edit "
29070  CLS
29080  KEY ON
29090 END
```

APPENDIX 7. Listing for PREPLOT.BAS (44 lines, 1024 bytes).

```
10 REM *** Prepare data for plot and fit programs***
20 PRINT "Chaining to plot and fit...."
30 PLTLABLY$=TITLE1OUT$
40 TITLE$=TITLE2OUT$
50 IF PLOTYFLAG=1 THEN ERASE PLOTY$
60 PLOTYFLAG=1
70 DIM PLOTY$(NAGESOUT,NYEARSOUT)
80 FOR I=1 TO NAGESOUT
90 FOR J=1 TO NYEARSOUT
100 PLOTY$(I,J)=MATRIXOUT$(I,J)
110 NEXT J,I
120 IF OUTFLAGC1=1 THEN ERASE C1MATRIX$
130 IF OUTFLAGC2=1 THEN ERASE C2MATRIX$
140 IF OUTFLAGC3=1 THEN ERASE C3MATRIX$
150 IF OUTFLAGC4=1 THEN ERASE C4MATRIX$
160 IF OUTFLAGC5=1 THEN ERASE C5MATRIX$
170 IF OUTFLAGC6=1 THEN ERASE C6MATRIX$
180 IF OUTFLAGC7=1 THEN ERASE C7MATRIX$
190 IF OUTFLAGC8=1 THEN ERASE C8MATRIX$
200 IF OUTFLAGC9=1 THEN ERASE C9MATRIX$
210 IF OUTFLAGC10=1 THEN ERASE C10MATRIX$
220 EFLAGC1=0
230 EFLAGC2=0
240 EFLAGC3=0
250 EFLAGC4=0
260 EFLAGC5=0
270 EFLAGC6=0
280 EFLAGC7=0
290 EFLAGC8=0
300 EFLAGC9=0
310 EFLAGC10=0
320 OUTFLAGC1=0
330 OUTFLAGC2=0
340 OUTFLAGC3=0
350 OUTFLAGC4=0
360 OUTFLAGC5=0
370 OUTFLAGC6=0
380 OUTFLAGC7=0
390 OUTFLAGC8=0
400 OUTFLAGC9=0
410 OUTFLAGC10=0
420 ERASE MATRIXOUT$
430 CALCFLAG=1
440 CHAIN "PLOT",,ALL
```

APPENDIX B. Listing for PLOT.BAS (638 lines, 20736 bytes without indentation).

```
10 '*****PLOT & FIT PROGRAM*****PROGRAMMER: MATHIEU JOLY*****
1000  DIM OFKEY$(4),NUMIMAGE(10),FITB#(3),FSTERR#(3)
1010  DEF FNMAX(X,Y)=(X+Y+ABS(X-Y))/2
1012  DEF FNLONG(Z)=LEN(STR$(Z))
1014  DEF FNTIC(X,$)=CINT(X*10^$)
1015  DEF FNSCALE(Z)=3-CINT(LOG(Z)/LOG(10))
1017  DEF FNLMAX(R,S)=FNMAX(FNLONG(R),FNLONG(S))
1018  DEF FNPLX(T)=60+(T-TICIOX)*PLTXSCALE
1019  DEF FNPLY(T)=154+(T-TICBOTY%)*PLTYSCALEY
1020  KEY OFF : SCREEN 0,0,0 : COLOR 7,0 : PLTITLE$ = TITLE2OUT$
1025    IF (FROMRESIDS) THEN  FROMRESIDS=0 : GOSUB 6000 ' => Fit Menu
1030    FOR I=1 TO 10 : KEY I,"" : NEXT
1035  PLTDELXTIC%=57 : PLTDELTICY%=-28
1037  GOSUB 1500
1040  REM ***PLOT MENU***
1045    ON ERROR GOTO 1300
1050  SCREEN 2,0 : CLS : PRINT "      >>>      PLOT  MENU       <<<"
1060  PRINT : PRINT "F1  : Automatic Scaling and PLOT (BEGIN WITH THIS OPTION)"
1070  PRINT : PRINT "F2  : PLOT again"
1080  PRINT : PRINT "F3  : Specify X Scale"
1090  PRINT : PRINT "F4  : Specify Y Scale"
1100  PRINT : PRINT "F5  : CHANGE Graph Title"
1110  PRINT : PRINT "F6  : CHANGE Axes Labels"
1120  PRINT : PRINT "F7  : Select AGES to be plotted"
1130  PRINT : PRINT "F9  : Return to MAIN PROGRAM"
1140  PRINT : PRINT "F10 : FIT a Curve"
1150  PRINT : PRINT "Press an F-key"
1160  GOSUB 1800    'Wait for F-key input.
1170    IF FKIN>7 THEN 1210
1180    ON FKIN GOSUB 1900,1910,4000,4510,4710,5000,1530
1190  GOTO 1050
1210    IF (FKIN=10) THEN GOSUB 6000  '*** FIT MENU
1220    IF (FKIN=9) THEN 10000 ' return to main pgm.
1240  GOTO 1050
1290  REM
1300  REM ***ERROR-HANDLING for PLOT Routines***
1310  SOUND 300,10 : CLS
1320  PRINT "Irrecoverable error in a PLOT routine ..."
1330  PRINT : PRINT "Error Code = "ERR,"in line # "ERL
1340    FOR I=1 TO 3000 : NEXT
1350  RESUME 1040
1390  REM
1400  REM  *** (ERROR ROUTINE ...)
1410  BEEP : CLS
1420  PRINT "*** The AUTOMATIC PLOTTING routine (F1 key) must be run BEFORE"
1430  PRINT "    the graph can be plotted using the F2 key.
1440  LOCATE 5,1 : PRINT "(The Auto. Plot computes certain values required by the actual Plot routine.)"
1450    FOR I=1 TO 3000 : NEXT
1460  RETURN 1040
1490  REM
1500  PLFROW=1
1510    IF NAGESOUT=1 THEN PLROWL=1 : RETURN
1520  PLROWL=NAGESOUT
1530  CLS :PRINT "The Input matrices have dimensions :  "INAGESOUT" x "INYEARSOUT
1535  FAGE=FAGEOUT
1540  PRINT :PRINT "The FIRST AGE in the matrices is : "IFAGE
1550  PLFAGE=FAGE+PLFROW-1 : PLABEL=FAGE+PLROWL-1
1560  PRINT :PRINT :PRINT "Ages currently selected :   FIRST AGE = "IPLFAGE
1570  PRINT TAB(30) "LAST AGE = "IPLABEL
1580  PRINT :PRINT "CHANGE these ages ? (Y/N)"
1590  K$=INKEY$ : IF K$="" THEN 1590
1600    IF (K$<>"y") AND (K$<>"Y") THEN 1670
```

```
1610  PRINT :INPUT "Select FIRST AGE ";PLFAGE
1620    IF (PLFAGE<FAGE) THEN BEEP : PRINT "Age TOO LOW ...": GOTO 1610
1630  PLFROW=PLFAGE-FAGE+1
1640  PRINT :INPUT "Select LAST AGE ";PLAGEL
1650    IF (PLAGEL>FAGE+NAGESOUT-1) THEN BEEP : PRINT "Age TOO HIGH ..." :GOTO 1640
1660  PLROWL=PLAGEL-FAGE+1 : PLTAUTO=0
1670  PARANGE$="(Ages  "+STR$(PLFAGE)+" to "+STR$(PLAGEL)+" )"
1680 RETURN
1790 REM  ***Get and validate an F-key input.
1795  BEEP
1800  K$=INKEY$ : IF K$="" THEN 1800
1810    IF (ASC(K$)>0) THEN 1800
1820  CODKIN=ASC(RIGHT$(K$,1
1825    IF (CODKIN<59) OR (CODKIN>68) THEN GOTO 1795
1840  FKIN=CODKIN-58
1850 RETURN
1860 REM
1870 REM  ***Automatic Plotting ...
1900 GOSUB 2000   '  Compute automatic scales.
1910 GOSUB 2900   '  Define F-keys and  PLOT.
1920 GOSUB 1800
1925    IF ((FKIN/3) AND (FKIN<9   OR (FKIN=10) THEN BEEP : GOTO 1920
1940    IF (FkIN=1) THEN RETURN
1950    IF (FkIN=2) THEN GOTO 6000  '*** FIT  MENU
1955    IF (FkIN=3) THEN GOSUB 9500 : GOTO 1920  '***SAVE SCREEN BUFFER
1960    IF (FkIN=9) THEN RETURN 10000  ' *** RET-MAIN
2000 REM *** AUTOMATIC SCALING  (ARGS :  PLOTX, PLOTY, PLOTSIZE)
2010 REM  1)  X SCALING :
2020  PLMIN=1E+10 : PLMAX=-PLMIN
2030    FOR I=PLFROW TO PLROWL
2035      FOR J=1 TO NYEARSOUT
2040        IF PLOTX@(I,J)<PLMIN THEN PLMIN=PLOTX@(I,J)
2050        IF PLOTX@(I,J)>PLMAX THEN PLMAX=PLOTX@(I,J)
2060    NEXT J,I
2070  PLTXMAX=PLMAX : PLTXMIN=PLMIN
2090  PLTNTICS=10
2100 GOSUB 2500
2110  TICX0X=BOTTIC : TICXTOPX=TOPTIC
2120  TICXDELTAX=TICDEL : PLTXFAC=SCALTIC
2132 REM
2135 REM  2)  Y SCALING :
2140  PLMIN=1E+10 : PLMAX=-PLMIN
2150    FOR I=PLFROW TO PLROWL
2155      FOR J=1 TO NYEARSOUT
2160        IF PLOTY@(I,J)<PLMIN THEN PLMIN=PLOTY@(I,J)
2170        IF PLOTY@(I,J)>PLMAX THEN PLMAX=PLOTY@(I,J)
2180    NEXT J,I
2190  PLTMINY=PLMIN : PLTMAXY=PLMAX
2210  PLTLABLY$=TITLE1OUT$
2220  PLTNTICS=5
2230 GOSUB 2500
2240  TICBOTY%=BOTTIC : TICTOPY%=TOPTIC
2250  TICDELTAY%=TICDEL : PLTFACY=SCALTIC
2260  PLTAUTO=-1
2270 RETURN
2490 REM  ***COMPUTE TIC-MARK VALUES FOR ANY AXIS
2500 REM
2520  SCALTIC=FNSCALE(FNMAX(ABS(PLMIN),PLMAX))
2530  BOTTIC=FNTIC(PLMIN,SCALTIC)
2540  TOPTIC=FNTIC(PLMAX,SCALTIC)
2550  TICDEL=(TOPTIC-BOTTIC)/PLTNTICS
2560    IF (TICDEL>INT(TICDEL)) THEN TICDEL=INT(TICDEL+1)
2570  TOPTIC=BOTTIC+PLTNTICS*TICDEL
2580 RETURN
2900  DFKMAX=4
```

```
2910  DFKEY$(1)="F1 : PLOT Menu"
2920  DFKEY$(2)="F2 : FIT Curve"
2925  DFKEY$(3)="F3 : BSAVE Screen"
2930  DFKEY$(4)="F9 : RETURN to MAIN"
2940  CLS
2980  REM
2990  REM *** PLOT ROUTINE  : for SCREEN DISPLAY.
3000  REM                    (Inputs : PLOTI$, PLOTY$, PLOTSIZE ).
3010    IF NOT (PLTAUTO) THEN 1400
3020  KPOSDEL=80\DFKNAI : KPOSCOL=1
3030    FOR I=1 TO DFKNAI
3040      LOCATE 25,KPOSCOL : PRINT DFKEY$(I);
3050      KPOSCOL=KPOSCOL+KPOSDEL
3060    NEXT
3070    IF BSAVEFLAG%=1 THEN BSAVEFLAG%=0 : RETURN
3080  GOSUB 3400
3090  LOCATE 24,1
3100  PRINT "Y min= ";PLTYMIN TAB(50) "X min= ";PLTXMIN;
3105  LINE (0,191)-(639,191)
3107    IF TITPLFLAG=1 GOTO 3120
3108  TITPLFLAG=1
3110    IF (LEN(PLTITLE$)<64)  THEN  TITPL$=PLTITLE$+"  Run 0"+STR$(RUNNUM) : GOTO 3120
3111    IF LEN(PLTITLE$)<73 THEN TITPL$=PLTITLE$ : GOTO 3120
3113  BEEP : LOCATE 23,1
3115  PRINT "***  Graph Title is TOO LONG ...   TRUNCATED ***
3117    FOR I=1 TO 1000 : NEXT
3119  TITPL$=LEFT$(PLTITLE$,72)
3120  LPL%=LEN(TITPL$)
3130  LOCATE 1,1 : PRINT TAB(44-LPL%\2) TITPL$
3140  LOCATE 23,1 : PRINT SPACE$(79);
3150  LOCATE 23,1 : PRINT TAB(42-LEN(PLTILABL$)\2) PLTILABL$;
3160  PRTICX%=48 : PRTICY%=161
3170  GOSUB 3500                    '  Draw  X - AXIS
3172  PRTICX%=19 : PRTICY%=151 : TICOLY%=1
3173  GOSUB 3700                    '  Draw  Y - AXIS
3180    IF (PLTMINY>=0) OR (PLTMAXY<=0) THEN 3240
3190      YPL=154-(TICBOTY%+PLTSCALEY)
3200      LINE (50,YPL)-(639,YPL)       '  Draw  X-AXIS in middle of graph
3240  GOSUB 3900  'Plot DATA points
3250  RETURN
3395  REM  ***DISPLAY the Y-LABEL.
3400  LPL%=LEN(PLTLABLY$)
3405    IF (LPL%>19) THEN 3440
3410  L1=11-LPL%\2
3420    FOR J=1 TO LPL%
3425      LOCATE L1,1 : PRINT MID$(PLTLABLY$,J,1)
3426      L1=L1+1
3430    NEXT
3435  RETURN
3440  LOCATE 23,1
3445  PRINT "*** Y-LABEL IS TOO LONG ; truncated to 19 columns ***" : BEEP
3450  PLTLABLY$=LEFT$(PLTLABLY$,19) : LPL%=19
3455  GOTO 3410
3490  REM  Display the X-axis and tics.
3500  LINE (60,150)-(630,150)
3510  TICVAL%=TIC1%
3520    WHILE TICVAL%<TICXTOP%
3523      XTIC%=PRTICX%+12
3526      LINE(XTIC%,154)-(XTIC%,150)
3530      GOSUB 3630
3540      TICVAL%=TICVAL%+TICXDELTA%
3550      PRTICX%=PRTICX%+PLTDELXTIC%
3560    WEND
3570  LINE (629,154)-(629,150)   'Display LAST TIC.
3580  PRTICX%=607 : GOSUB 3630
3590  PLTXSCALE=570/(TICXTOP%-TIC1%)
3591  MX=FNPLX(PLTXMAX*10^PLTXFAC)
```

```
3593  LINE (RX-2,158)-(RX,152)  ' display a triangle at true DATA minimum
3595  LINE -(RX+2,158)          ' on X axis.
3600 RETURN
3610 REM
3620 REM *** PRINT A TIC at the proper location .
3630  LOCATE 1,1
3632    IF TICVALX<100 THEN PRINT TICVALX ELSE PRINT MID$(STR$(TICVALX),2)
3640    GET(0,0)-(31,7),NUMIMAGE
3650    PUT(0,0),NUMIMAGE
3660    PUT (PRTICX%,PRTICY%),NUMIMAGE
3670 RETURN
3680 REM
3690 REM  ***Display  Y axis  and tics.
3700  LINE(50,9)-(50,154)    ' Axis at LEFT
3710  LINE(639,154)-(639,9)  ' Axis at  RIGHT.
3720  TICVALX=TICBOTY%
3730    WHILE TICVALX<=TICTOPY%
3740      TY%=PRTICY%+3
3750      LINE (50,TY%)-(54,TY%)
3760      LINE (635,TY%)-(639,TY%)
3770      GOSUB 3630
3780      TICVALX=TICVALX+TICDELTAY%
3790      PRTICY%=PRTICY%+PLTDELTICY%
3800    WEND
3810  PLTSCALEY=-140/(TICTOPY%-TICBOTY%)
3820  HY=FNPLY(PLTMAXY*10^PLTFACY)
3830  LINE (639,HY-2)-(631,HY),TICOLY%
3840  LINE -(639,HY+2),TICOLY%
3850 RETURN
3860 REM
3890 REM  ***PLOT DATA POINTS
3900    FOR I=PLFROW TO PLROWL
3905      FOR J=1 TO NVEARSOUT
3910        IPL=FNPLX(PLOTX%(I,J)*10^PLTXFAC)
3920        VPL=FNPLY(PLOTY%(I,J)*10^PLTFACY)
3930        CIRCLE(IPL,VPL),2,,,,.8
3940    NEXT J,I
3950    IF PLFROW<>PLROWL THEN RETURN
3960  LINE ((IPL-3),(VPL-3))-((IPL+3),(VPL+3))
3965  LINE ((IPL-3),(VPL+3))-((IPL+3),(VPL-3))
3970 RETURN
3990 REM  ***SPECIFY X - SCALE
4000  CLS : PRINT ")))  X - SCALE  SPECIFICATION"
4010  LOCATE 4,1 : PRINT "Current  Values :
4020  LOCATE 6,1 : PRINT "min(x) =";PLTXMIN,"Max(x) =";PLTXMAX
4030  LOCATE 8,1 : PRINT "Scaling factor = 10 ^^";PLTXFAC
4040  LOCATE 10,1: PRINT "First Tick =";TICX0%  TAB(40) "Last Tick =";TICXTOP%
4050  LOCATE 12,1: PRINT "Tick Increment =";TICXDELTA%
4060  LOCATE 14,1: PRINT ")))  Enter NEW parameters, in BARE UNITS as data :"
4070  LOCATE 16,1: INPUT "Enter 1st Tick ";TOB%
4080    IF (TOB%<=PLTXMIN) THEN 4110
4090  BEEP : LOCATE 25,1
4100  PRINT "*** 1st tick should be <= min(x) *** " : GOTO 4070
4110  BXMAX=PLTXMAX
4115  LOCATE 25,1 : PRINT $PACE$(79)
4120  COLB%=1 : BXNTICS=10 : GOSUB 4270
4130  TICX0%=BX0% : TICXTOP%=BXTOP%
4140  TICXDELTA%=BXDEL% : PLTXFAC=BXSCALE
4150  LOCATE 19,1 : PRINT TAB(40) "NEW  X - AXIS :"
4160  LOCATE 20,1 : PRINT $PACE$(79)
4170  LOCATE 21,1 : PRINT $PACE$(80)
4180  PRTICX%=48 : PRTICY%=161 : GOSUB 3500
4182  HX=FNPLX(PLTXMIN*10^PLTXFAC)
4184  GOSUB 3593  ' Draw a triangle at data minimum on X-axis
4190  LOCATE 23,1 : PRINT "Is this correct ? (Space = OK)";
4200    A$=INKEY$ : IF A$="" THEN 4200
```

```
4210    IF (A$=CHR$(32)) THEN RETURN
4220  LOCATE 23,1: PRINT SPACE$(79);
4230  LOCATE 24,1: PRINT SPACE$(79);
4240  LOCATE 25,1: PRINT SPACE$(79); : GOTO 4070
4250 REM
4260 REM  ***Compute new tic-values and scale factor (for X or Y axis).
4270  LOCATE 18,COL$1 : INPUT "Enter new INCREMENT ";DEL$1
4280  TOP$1=TOS1+SINTICS*DEL$1
4290    IF TOP$1)=SINA1 THEN 4330
4300  BEEP : LOCATE 25,1
4310  PRINT "*** LAST TICK IS TOO SMALL : ";TOP$1 TAB(50) "Increase INCREMENT ***";
4320  LOCATE 18,1 : PRINT SPACE$(79) : GOTO 4270
4330  SISCALE=0
4340  LMAIS1=FNLMAI(TOS1,TOP$1)
4350    IF (DEL$1<>INT(DEL$1)) OR (TOS1<>INT(TOS1)) THEN 4370
4360    IF (LMAIS1=<5) AND (FNLONG(TOS1)<5) THEN 4440
4370  SISCALE=FNSCALE(FNMAI(ABS(TOS1),TOP$1))
4380  SIO1=FNTIC(TOS1,SISCALE)
4390    IF FNLONG(SIO1)=5 THEN SISCALE=SISCALE-1
4400  BEEP : LOCATE 24,1
4410  PRINT "*** LARGEST TICK HAS TOO FEW/MANY DIGITS ... TICK VALUES ARE SCALED DOWN :";
4420  LOCATE 25,1 : PRINT SPACE$(79);
4430  LOCATE 25,1 : PRINT "*** NEW SCALING FACTOR = 10 ^";SISCALE;
4440  SIDEL1=FNTIC(DEL$1,SISCALE)
4450  SIO1=FNTIC(TOS1,SISCALE)
4460  SITOP1=SIO1+SINTICS*SIDEL1
4470    IF (SITOP1)=SINA1*10^SISCALE) THEN RETURN
4480  LOCATE 24,1 : PRINT "AFTER SCALING & ROUNDING ... "+SPACE$(49) : GOTO 430
4490 REM
4500 REM  ***SPECIFY Y - SCALE
4510  CLS : PRINT TAB(20) ")))  Y SCALE  SPECIFICATION  (((";
4520  LOCATE 4,30: PRINT "Current  Values :
4530  LOCATE 6,20: PRINT "Min(Y) =";PLTMIN1,"Max(Y) =";PLTMAX1
4540  LOCATE 8,20: PRINT "Scaling factor = 10 ^";PLTFACY
4550  LOCATE 10,20: PRINT "First Tick =";TICBOTY1  TAB(40) "Last Tick =";TICTOPY1
4560  LOCATE 12,20: PRINT "Tick Increment =";TICDELTAY1
4570  LOCATE 14,20: PRINT ")))  Enter NEW parameters, in SAME UNITS as data !";
4580  LOCATE 16,20: INPUT "Enter 1st Tick ";TOS1
4590    IF (TOS1<=PLTMIN1) THEN 4620
4600  BEEP : LOCATE 25,1 : PRINT SPACE$(79); : LOCATE 25,1
4610  PRINT "*** 1st tick should be <= min(Y) *** "; : GOTO 4580
4620  SIMAI=PLTMAX1
4625  LOCATE 25,1 : PRINT SPACE$(79);
4630  COL$1=20 : SINTICS=5 : GOSUB 4270
4640  TICBOTY2=SIO1 : TICTOPY1=SITOP1
4650  TICDELTAY2=SIDEL1 : PLTFACY=SISCALE
4660  LOCATE 1,1 : PRINT  "NEW  Y - AXIS :"
4670  PRTICX1=19 : PRTICY2=151 : GOSUB 3700
4672  MY=FNPLY(PLTMIN1+10^PLTFACY) : MINSY=MY
4674 GOSUB 3830  ' Draw a triangle at data minimum on Y-axis (at left).
4680  LOCATE 23,1 : PRINT "Is this correct ? (Space = OK)";
4690  A$=INKEY$ : IF A$="" THEN  4690
4700    IF (A$=" ") THEN RETURN
4705  PRTICX1=19 : PRTICY2=151 : TICDLY2=0 : GOSUB 3700
4707  MY=MINSY : GOSUB 3830 : TICDLY2=1 : GOTO 4580
4710 REM  ***
4720  CLS : PRINT ")))  CHANGE THE GRAPH TITLE  (((";
4730  LOCATE 4,1 : PRINT "Current title =";TITPL$
4740  LOCATE 7,1 : PRINT "NOTE :  Graph Title should be LESS THAN 70 chars long, if the 'RUN 0 ' is to be appended."
4750 GOSUB 4810
4760    IF A$=" " THEN RETURN
4770  LC=12 : NCHAR=69 : A$="Enter New Title :"
4780 GOSUB 4850 : PLTITLE$=R$ : TITPL$=R$ : TITPLFLAG=0
4790 RETURN
4800 REM
4810  PRINT "CHANGE this title ?  (Space = NO Change) ";
4820  A$=INKEY$ : IF A$="" THEN 4820
4830 RETURN
```

```
4840 REM
4850  LOCATE LC,1 : PRINT R$
4860  PRINT "|"+STRING$((NCHAR-2),"-")+"|"
4870  LINE INPUT R$
4880  R$=LEFT$(R$,NCHAR) : RETURN
4890 REM
4990 REM  ***(F6 KEY) ...
5000  CLS : PRINT ")))  Change  the X-AXIS LABEL  ((("
5010  LOCATE 4,1 : PRINT "Current LABEL = "|PLTXLABL$
5020  LOCATE 7,1 : PRINT "NOTE :  X-Label should be LESS THAN 70 chars long "
5030 GOSUB 4810
5040    IF R$=" " THEN 5100
5050  LC=12: NCHAR=69 : R$="Enter NEW LABEL :"
5060 GOSUB 4850 : PLTXLABL$=R$
5070 REM
5100  CLS : PRINT ")))  Change  the Y-AXIS LABEL  ((("
5110  LOCATE 4,1 : PRINT "Current LABEL = "|PLTLABLY$
5120  LOCATE 7,1 : PRINT "NOTE :  Y-Label should be LESS THAN 20 chars long"
5130  GOSUB 4810
5140    IF R$=" " THEN RETURN
5150  LC=12: NCHAR=19 : R$="Enter NEW LABEL :"
5160 GOSUB 4850 : PLTLABLY$=R$
5170 RETURN
5180 REM
5990 REM
5990 REM  ***  CURVE FITTING ROUTINES ...
6000    ON ERROR GOTO 10680 ' ERROR-HANDLING for  FIT Section.
6010  FITCOMP0=0 : FITSLEVEL$="5" : SCREEN 0,0,0
6020  CLS: PRINT "     )))  FIT  MENU  (((
6030  PRINT :PRINT "F1  :  Linear Regress.on  (Simple Least Squares)"
6040  PRINT :PRINT "F2  :  Linear Regression  (Through Origin)"
6050  PRINT :PRINT "F3  :  Linear Regression  (Geometric Mean)"
6060  PRINT :PRINT "F4  :  Quadratic Regression "
6070  PRINT :PRINT "F7  :  Change significance level <- Current level = .0"|FITSLEVEL$
6080  PRINT :PRINT "F9  :  Return to MAIN Pgm."
6090  PRINT :PRINT "F10 :  Return to PLOT Menu"
6100  PRINT :PRINT "Press an F-key"
6110 GOSUB 1004
6120    IF FKIN<5 THEN GOSUB 6180 : GOTO 6020
6130    IF (FKIN=5) OR (FKIN=6) OR (FKIN=8) THEN BEEP : GOTO 6110
6140    IF (FKIN=7) THEN GOSUB 8300 : GOTO 6020
6150    IF (FKIN=9) THEN RETURN 10000
6160    IF (FKIN=10) THEN RETURN 1050
6170 REM
6180 REM *** Initial computations (sums, means ...)
6200    IF (FITCOMP0) THEN  6460
6208  CLS
6210  PRINT "Input matrices range over years :"|FYEAROUT|" to"|(FYEAROUT+NYEARSOUT-1)
6212  PRINT :PRINT "Do you wish to EXCLUDE A FEW YEARS from the Regression fitting ? Y/N"
6214  K$=INKEY$ : IF K$="" THEN 6214
6216    IF (K$="N") OR (K$="n") THEN FTNY=NYEARSOUT : GOTO 6224
6218  PRINT :INPUT "Enter LAST YEAR for computations"|LYRFT
6222  FTNY=LYRFT-FYEAROUT+1
6224  NTOT0=(PLRGUL-PLFROM+1)*FTNY
6226    IF (NTOT0<3) THEN 6235
6228  BEEP : PRINT "*** There should be AT LEAST 4 POINTS included in the computations"
6230  PRINT :PRINT "INCREASE the LAST YEAR ..." : GOTO 6218
6235  CLS : PRINT "I'M  THINKING  ...  PLEASE BE PATIENT !"
6240  FSI0=0 : FSY0=0 : FSI20=0 : FSY20=0 : FSIY0=0
6250    FOR I=PLFROM TO PLRGNL   '  Compute 'raw' sums (x,x2,y,y2)
6260      FOR J=1 TO FTNY
6270        FSI0=FSI0+PLOTI0(I,J)
6280        FSI20=FSI20+PLOTI0(I,J)*PLOTI0(I,J)
6290        FSY0=FSY0+PLOTY0(I,J)
6300        FSY20=FSY20+PLOTY0(I,J)*PLOTY0(I,J)
6310        FSIY0=FSIY0+PLOTI0(I,J)*PLOTY0(I,J)
6320      NEXT
6330    NEXT
```

```
6340  FTXMX8=FSX8/WTOT8
6350  FTMX20=FSX20/WTOT8
6360  FTMY8=FSY8/WTOT8
6410  FTSX8=FSX20-(FSX8*FSX8)/WTOT8
6420  FTSY8=FSY20-(FSY8*FSY8)/WTOT8
6430  FTXY8=FSXY8-(FSX8*FSY8/WTOT8)
6440  FITCOMP0=-1  ' INITIAL Computations done.
6450 REM  Print heading and select  proper fitting routine.
6460  CLS: SCREEN 0,0,0 : LOCATE 23,1 : COLOR 0,7
6470  PRINT "F1  PLOT Curve" TAB(60) "F10  ) FIT Menu";
6480  COLOR 7,0
6490  LOCATE 1,25 : PRINT "MODEL  :  Y = "
6500  LOCATE 3,47 : PRINT "t  Values"
6510  LOCATE 4,1 : PRINT "Degree   Coefficient" TAB(25) "Std. Error" TAB(40) "Computed" TAB(53) "Table     (";FITSLEVEL8;"%)"
6520    ON FXIN GOTO 6540,7400,7530,7600
6530 REM
6540 REM  *** Model  :  Y = A + Bt
6550  LOCATE 1,40 : PRINT "B0 + B1 t + e"
6560  FITB8(1)=FTXY58/FTSX18
6570  FITB8(0)=FTMY8-FITB8(1)*FTXM8
6580  FITB8(2)=0
6590  FTREGSS8=FTXY58*FITB8(1)  ' SS(b) // b0)
6600  FTSSRES8=FTSY8-FTREGSS8  ' Residual  SS
6610  FTRESDF=WTOT8-2
6618  V88FT8=FTSSRES8/FTRESDF
6620  STDE=FT8=SQR(V88FT8)
6630  FSTERR8(0)=SQR(FSX20/(WTOT8*FTSX18)*V88FT8)
6640  FSTERR8(1)=SQR(V88FT8/FTSX18)
6650 GOSUB 6870  ' Print coefficients and t-values...
6660 GOSUB '6BC  ' Print beginning of Anova table
6670  FTSOURCE9="B1 , b0" : FTDF=1
6680  FTSS8=FTREGSS8
6690  FCOMP=FTRESDF*FTREGSS8/FTSSRES8
6700 GOSUB 7080  ' Print 'regression' line of Anova table
6710  FTSOURCE9="Residual" : FTDF=FTRESDF
6720  FTSS8=FTSSRES8 : FCOMP=0
6730 GOSUB 7080  ' Print 'residual' line of table
6740  PRINT STRING8(79,"-")
6750  PRINT "Total" TAB(15) WTOT8;
6760  PRINT TAB(20) USING "88.88888^^^^";FSY20
6770    IF FXIN=2 THEN R2FT8=FTREGSS8/FSY20 ELSE R2FT8=FTREGSS8/FTSY18
6780  PRINT :PRINT USING "R squared =  0.888888";R2FT8
6790  PRINT
6795    IF FXIN=4 THEN RETURN  ' ... to Quadratic fit.
6800  PRINT USING"Correlation Matrix :   I     0.88888";FTXY58/SQR(FTSX18*FTSY18)
6810  PRINT TAB(30);"I"
6820 GOSUB 1800
6830    IF FXIN=1 THEN 6900  'eeee  PLOT line
6840    IF FXIN=10 THEN RETURN
6850  BEEP : GOTO 6830   ' Invalid F-key entered
6860 REM  DISPLAY  first table (coefficients)
6870  FLN=5 : FTDFNUM=1
6880 GOSUB 7130  ' get F-table value from diskette file
6890  LOCATE FLN,1
6900    FOR I=0 TO 2
6910      IF FITB8(I)=0 THEN 6970
6920      FCOMPT=ABS(FITB8(I)/FSTERR8(I))
6930      PRINT " ";I
6940      LOCATE FLN,10 : PRINT USING "88.88888^^^^   ";FITB8(I),FSTERR8(I);
6950      LOCATE FLN,40 : PRINT USING "8888.888    ";FCOMPT,xBR(FTABLE7)
6960      FLN=FLN+1
6970    NEXT
6980 RETURN
6990 REM  ***Print beginning of ANOVA table.
7000  LOCATE 9,30 : PRINT "A N O V A"
7010  PRINT "Source" TAB(16) "DF" TAB(25) "SS" TAB(39) "MS" TAB(50) "Computed F" TAB(68) "Table F"
```

```
7020    FLN=11 : IF FITB@(0)=0 THEN RETURN
7030  PRINT "Mean" TAB(17) "1";
7040  PRINT TAB(20) USING "@@.@@@@^^^^" ;((FSY@+FSY@)/NTOT@)
7050  FLN=FLN+1
7060 RETURN
7070 REM  ***Print a line of the ANOVA table
7080  PRINT FTSOURCE$ TAB(14) USING "@@@@" ;FTDF;
7090  PRINT TAB(20) USING "@@.@@@@^^^^    ";FTSS@,(FTSS@/FTDF);
7100    IF FCOMP=0 THEN PRINT : RETURN
7110  PRINT TAB(50) USING "@@@@@.@@       ";FCOMP,FTABLEF
7120 RETURN
7130 REM  ***READ F-TABLE Value for (ftdfnum,ftresdf;fitslevel$)
7140  TABFILE$="FTAB"+RIGHT$(STR$(FTDFNUM),1)+"DF.0"+FITSLEVEL$
7150    OPEN TABFILE$ FOR INPUT AS #3
7160      DRES=FTRESDF
7170        IF DRES>30 THEN 7210
7180      DFN=DRES : GOSUB 7300
7190      FTABLEF=FVAL
7200    CLOSE #3 : RETURN
7210        IF DRES<40 THEN DFN=30 : DFG=30 : DFD=40 : GOTO 7250
7220        IF DRES<60 THEN DFN=31 : DFG=40 : DFD=60 : GOTO 7250
7230        IF DRES<120 THEN DFN=32 : DFG=60 : DFD=120 : GOTO 7250
7240      DFN=33 : DFG=120 : DFD=1000
7250      GOSUB 7300
7260      TF1=FVAL
7270      INPUT#3,TF2  ' F-value for df. at RIGHT of interval.
7280      FTABLEF=TF1+((1/DRES)-(1/DFG))/((1/DFD)-(1/DFG))*(TF2-TF1) 'Interpolation
7290    CLOSE #3 : RETURN
7300 REM *** get the f-value
7310    FOR I=1 TO DFN
7320      INPUT#3,FVAL
7330    NEXT
7340 RETURN
7350 REM *** Regression line through the ORIGIN
7400  LOCATE 1,40 : PRINT "B I + e  -- (Line through the ORIGIN)"
7410  FITB@(0)=0 : FITB@(2)=0
7420  FITB@(1)=FSIY@/FSI2@
7430  FTREGSS@=FSIY@*FITB@(1)
7440  FTSSRES@=FSY2@-FTREGSS@
7450  FTRESDF=NTOT@-1
7460  VARFT@=FTSSRES@/FTRESDF
7470  FSTERR@(1)=SQR(VARFT@/FSI2@)
7480 GOSUB 6870
7490 GOSUB 7000
7500  FTSOURCE$="Regression" : FTDF=1
7510 GOTO 6680
7520 REM  *** GEOMETRIC MEAN linear regression
7530  LOCATE 1,15 : PRINT "MODEL  :  Y = B0 * B1 I * e  -- B1 = GEOMETRIC MEAN Estimate"
7540  FITB@(1)=SGN(FTIY5@)*SQR(FTSYY@/FTSII@)
7550 GOTO 6570  ' ** REST of computations are the same as Least Squares Linear
7560 REM
7590 REM  ***Quadratic curve fitting
7600  LOCATE 1,40 : PRINT "B0 + B1 I + B2 I^2 + e"
7610  LOCATE 6,1 : PRINT "COMPUTING  MORE  SUMS ..."
7620  FSSIS@@=0 : FSSI12@=0 : FSSI2Y@=0 : FSII3@=0
7630    FOR I=PLFROW TO PLROWL
7640      FOR J=1 TO FTWY
7650        DIFFT@=PLOTI@(I,J)*PLOTI@(I,J)-FTMNI2@
7660        FSSIS@@=FSSIS@@+DIFFT@*DIFFT@
7670        FSSI2Y@=FSSI2Y@+DIFFT@*PLOTY@(I,J)
7680        FSSI12@=FSSI12@+DIFFT@*PLOTI@(I,J)
7685        FSII3@=FSII3@+PLOTI@(I,J)*PLOTI@(I,J)*PLOTI@(I,J)
7690      NEXT
7700    NEXT
7705  FSI4@=FSSIS@@+NTOT@*FTMNI2@*FTMNI2@
7710  DETFT@=FTSII@*FSSIS@@-FSSI12@*FSSI12@
```

```
7720  LOCATE 6,1 : PRINT SPACE$(40)
7730  FITB0(1)=(FTIY$0*FSSIS00-FSSIZY0*FSSI120)/DETFT0
7740  FITB0(2)=(FSSIZY0*FTSII0-FTIY$0*FSSI120)/DETFT0
7750  FITB0(0)=FTMNY0-FITB0(1)*FTIMN0-FITB0(2)*FTMNI20
7760  FTRE6S$0=FITB0(1)*FTIY$0+FITB0(2)*FSSIZY0
7770  FTSSRE$0=FTSYY0-FTRE6S$0 : FTRESDF=NTOT0-3
7780  VARFT0=ABS(FTSSRE$0/FTRESDF)
7782  DIM MVAR0(3,3)
7784  MVAR0(0,0)=FSI40*FSI20-FSI30*FSI30
7786  MVAR0(0,1)=FSI0*FSI40-FSI20*FSI30
7788  MVAR0(0,2)=FSI0*FSI30-FSI20*FSI20
7790  MVAR0(1,1)=NTOT0*FSSIS00
7792  MVAR0(1,2)=NTOT0*FSI30-FSI0*FSI20
7794  MVAR0(2,2)=NTOT0*FTSII0
7796  MFAC00=NTOT0*FSI20*FSI40+2*FSI0*FSI20*FSI30
7798  MFAC00=MFAC00-(NTOT0*FSI30*FSI30+FSI0*FSI0*FSI40+FSI20^3)
7800  MFAC00=VARFT0/MFAC00
7805    FOR I=0 TO 2
7810      FSTERR0(I)=SQR(MVAR0(I,I)*MFAC00)
7820    NEXT
7830 GOSUB 6870
7840 GOSUB 7000
7850  FTSOURCE$="b1 / b0" : FTDF=1
7860  FTSS0=FTIY$0*FTIY$0/FTSII0
7870  FCOMP=FTSS0/VARFT0
7900 GOSUB 7080
7920  FTSOURCE$="b2 /(b1,b0)"
7930  FTSS0=FTRE6S$0-FTSS0
7940  FCOMP=FTSS0/VARFT0
7950 GOSUB 7080
7960 FTDFNUM=2 : GOSUB 7140  ' READ F-TABLE FILE FOR F(2,N-3)
7970  FTDF=2 : FTSS0=FTRE6S$0
7980  FCOMP=FTSS0/(VARFT0*2)
7990 FTSOURCE$="(b1,b2) /b0" : GOSUB 7080
8000 GOSUB 6710
8010  LOCATE 21,1 : PRINT "Correlation matrix : Y   1";
8020  PRINT USING "   0.0000";FTIY$0/SQR(FTSII0*FTSYY0),FSSIZY0/SQR(FSSIS00*FTSYY0)
8030  RC12FT0=FSSI120/SQR(FTSII0*FSSIS00)
8040  PRINT "of variables" TAB(22) "I       1";
8045  PRINT TAB(39) USING "0.0000";RC12FT0
8050  PRINT TAB(22) "I^2" TAB(39) "1"
8060  DETCM0=1-RC12FT0*RC12FT0
8065  LOCATE 20,62 : PRINT "COLLINEARITY"
8070  LOCATE 21,60 : PRINT USING "Determinant = 0.0000";DETCM0;
8080  LOCATE 23,60 : PRINT USING "Sq. Root = 0.0000";SQR(DETCM0);
8100 GOSUB 1860
8110    IF FKIN=1 THEN 8150
8120    IF FKIN=10 THEN ERASE MVAR0 : RETURN
8130  BEEP : GOTO 8100   '  Invalid F-key entered
8140 REM  ** define fitting function and plot it...
8150  DEF FNYHAT(I)=FITB0(0)+I*(FITB0(1)+FITB0(2)*I)
8159  ERASE MVAR0
8160 GOTO 8910
8300  CLS: PRINT "CURRENT SIGNIFICANCE LEVEL = "+FITSLEVEL$+"%"
8310  PRINT :PRINT "CHANGE THIS LEVEL ? (Y/N)"
8320  R$=INKEY$ : IF R$="" THEN 8320
8330    IF (R$<>"y") AND (R$<>"Y") THEN RETURN
8340  FITSLEVEL$=RIGHT$(STR$(6-VAL(FITSLEVEL$)),1)
8350 RETURN
8900  DEF FNYHAT(I)=FITB0(0)+FITB0(1)*I
8910  FTNINT=57 : GOSUB 9000
8920    IF FKIN=10 THEN RETURN 6160
8930    IF FKIN=5 THEN GOSUB 9300 '***  chain the RESIDUALS routines
8935    IF FKIN=3 THEN GOSUB 9500 : GOSUB 1800 : GOTO 8920
8940 RETURN  ' fkin=1 : =) FIT menu
```

```
8990 REM  ***PLOT the fitted curve / line.
9000  DFKMAX=4
9010  DFKEY$(1)="F1 : FIT Menu"
9015  DFKEY$(2)="F3 : BSAVE Screen"
9020  DFKEY$(3)="F5 : RESIDUALS Menu"
9030  DFKEY$(4)="F10 : PLOT Menu"
9040  SCREEN 2,0 : GOSUB 3000 : ' Plot axes & data
9050  XDELFT=(TICITOP1-TICI01)/FTNINT
9060  ILFT=TICI01
9070    WHILE ILFT<TICITOP1
9080      YLFT=FNYHAT(ILFT/10^PLTIFAC)+10^PLTFACY
9090      IRIT=ILFT+IDELFT
9100      YRIT=FNYHAT(IRIT/10^PLTIFAC)+10^PLTFACY
9110        IF (YLFT<TICIBOTY1) OR (YLFT)TICTOPY1) THEN 9130
9115        IF (YRIT<TICIBOTY1) OR (YRIT)TICTOPY1) THEN 9130
9120      LINE (FNPLX(ILFT),FNPLY(YLFT))-(FNPLX(IRIT),FNPLY(YRIT))
9130      ILFT=ILFT+IDELFT
9140    WEND
9150 GOSUB 1800
9160    IF (FKIN<>1) AND (FKIN<>3) AND (FKIN<>5) AND (FKIN<>10) THEN BEEP : GOTO 9150
9170 RETURN
9300 CHAIN "resids",100,ALL
9500 REM ***BSAVE the screen buffer
9510  BSAVEFLAG1=1
9520  LOCATE 25,1 : PRINT SPACE$(79);
9530  LINE (0,191)-(639,191),0
9540  LOCATE 22,1 : INPUT "Enter filename:   ",BSAVEFN$
9550  LOCATE 22,1 : PRINT SPACE$(40);
9560  BSAVEFN$="B:"+BSAVEFN$
9570  DEF SEG=&HB800
9580    BSAVE BSAVEFN$,0,&H4000 : DEF SEG
9590 GOSUB 3020
9600  LINE (0,191)-(639,191)
9610 RETURN
10000 REM *** RETURN TO MAIN PROGRAM ...
10010  CLS : ON ERROR GOTO 0
10015  PRINT "CHAINING BACK TO MAIN PGM (COHORT)..."
10020  ERASE PLOTY$,FITB$,FSTERR$,DFKEY$,MUKIMAGE
10030  PLOTYFLAG=0 : TITPLFLAG=0
10040 CHAIN "cohort",29200,ALL
10050 END
10060  PRINT "***  ERROR  in a FIT routine ..." :BEEP
10090  PRINT "Error #  "ERR,"in line #  "ERL
10095    FOR I=1 TO 3000 : NEXT
10100 RESUME 6020
```

APPENDIX 9. Table showing the contents of the four diskette files that generate the $F_{\nu_1,\nu_2,1-\alpha}$ ($=(t_{\nu_2,1-\alpha/2})^2$ when $\nu_1=1$) values for the regression analysis in PLOT.BAS. The first column shows values for ν_2 (interpolation is done for df>30). The names of the files are displayed at the top of the remaining four columns. Of these columns, the first contains F values for $\nu_1=1$, $\alpha=.05$; the second $\nu_1=2$, $\alpha=.05$; the third $\nu_1=1$, $\alpha=.01$; the fourth $\nu_1=2$, $\alpha=.01$. Numbers are from Draper and Smith (1981).

DF	FTAB1DF.05	FTAB2DF.05	FTAB1DF.01	FTAB2DF.01
1	161.40	199.50	4052.00	4999.50
2	18.51	19.00	98.50	99.00
3	10.13	9.55	34.12	30.82
4	7.71	6.94	21.20	18.00
5	6.61	5.79	16.26	13.27
6	5.99	5.14	13.75	10.92
7	5.59	4.74	12.25	9.55
8	5.32	4.46	11.26	8.65
9	5.12	4.26	10.56	8.02
10	4.96	4.10	10.04	7.56
11	4.84	3.98	9.65	7.21
12	4.75	3.89	9.33	6.93
13	4.67	3.81	9.07	6.70
14	4.60	3.74	8.86	6.51
15	4.54	3.68	8.68	6.36
16	4.49	3.63	8.53	6.23
17	4.45	3.59	8.40	6.11
18	4.41	3.55	8.29	6.01
19	4.38	3.52	8.18	5.93
20	4.35	3.49	8.10	5.85
21	4.32	3.47	8.02	5.78
22	4.30	3.44	7.95	5.72
23	4.28	3.42	7.88	5.66
24	4.26	3.40	7.82	5.61
25	4.24	3.39	7.77	5.57
26	4.23	3.37	7.72	5.53
27	4.21	3.35	7.68	5.49
28	4.20	3.34	7.64	5.45
29	4.18	3.33	7.60	5.42
30	4.17	3.32	7.56	5.39
40	4.08	3.23	7.31	5.18
60	4.00	3.15	7.08	4.98
120	3.92	3.07	6.85	4.79
∞	3.84	3.00	6.63	4.61

APPENDIX 10. Listing for RESIDS.BAS (306 lines, 9742 bytes without indentation).

```
 10 '*****RESIDUALS PROGRAM*****PROGRAMMER: MATHIEU JOLY*****
 100  NAGESRES=PLROW-PLFROW : NYEARSRES=NYEARSOUT-1
 110  FAGERES=PLFAGE : FYEARES=FYEAROUT-1900
 120    IF FITD0(2)=0 THEN DEF FNYHAT(I)=FITD0(0)+FITD0(1)*I ELSE DEF FNYHAT(I)=FITD0(0)+I*(FITD0(1)+I*FITD0(2))
 130  DIM RESHAT(NAGESRES,NYEARSRES), RESI(NAGESRES,NYEARSRES)
 140    FOR I=0 TO NAGESRES
 150      FOR J=0 TO NYEARSRES
 160        II=PLFROW+I
 170        RESHAT(I,J)=PLOTY0(II,J+1)-FNYHAT(PLOTI0(II,J+1))
 180    NEXT J,I
 1010  DEF FNMAI(X,Y)=(X+Y+ABS(X-Y))/2
 1012  DEF FNLONG(Z)=LEN(STR$(Z))
 1014  DEF FNTIC(X,S)=CINT(X*10^S)
 1015  DEF FNSCALE(Z)=3-CINT(LOG(Z)/LOG(10))
 1017  DEF FNLMAX(R,S)=FNMAI(FNLONG(R),FNLONG(S))
 1018  DEF FNPLI(T)=60+(T-RSI01)*RESISCALE
 1019  DEF FNPLY(T)=154+(T-RSI0TY1)*RESCALEY
 1035  PLTDELTICI=57 : PLTDELTICYI=-28
 1045    ON ERROR GOTO 1300
 1050  SCREEN 0,0,0 : CLS : PRINT ">>>       RESIDUALS  MENU       <<<"
 1060  PRINT :PRINT "F1  : PLOT against I"
 1070  PRINT :PRINT "F2  : PLOT against YEARS"
 1080  PRINT :PRINT "F3  : PLOT against OBSERVED Y"
 1090  PRINT :PRINT "F4  : PLOT against PREDICTED Y"
 1120  PRINT :PRINT "F7  : Display on SCREEN"
 1130  PRINT :PRINT "F8  : PRINT Residuals"
 1140  PRINT :PRINT "F10 : <<) FIT Menu"
 1150  PRINT :PRINT "Press an F-key"
 1160 GOSUB 1800   ' Wait for F-key input.
 1170    IF FKIN>4 THEN 1210
 1175  SCREEN 2,0
 1180    ON FKIN GOSUB 4000,4070,4200,4300
 1190 GOTO 1050
 1210    IF (FKIN=10) THEN 8000  '<<<< RETURN TO FIT
 1220    IF (FKIN=5) OR (FKIN=6) OR (FKIN=9) THEN 1160
 1230    ON (FKIN-6) GOSUB 5000,6000
 1240 GOTO 1050
 1290 REM <<< ERROR-HANDLING for Residuals section.
 1300  SOUND 300,10 : CLS
 1310  PRINT "IRRECOVERABLE ERROR  in a RESIDUALS Routine..."
 1320  PRINT :PRINT "ERROR # ";ERR,"in line #";ERL
 1330    FOR I=1 TO 3000 : NEXT
 1340 RESUME 1045
 1790 REM  >>>  Get and validate an F-key input.
 1795  BEEP
 1800  K$=INKEY$ : IF K$="" THEN 1800
 1810    IF (ASC(K$)>0) THEN 1800
 1820  CODKIN=ASC(RIGHT$(K$,1))
 1830    IF (CODKIN<59) OR (CODKIN>68) THEN GOTO 1795
 1840  FKIN=CODKIN-58
 1850 RETURN
 1880 REM
 1890 REM  <<<Autoscatic Plotting ...
 1900 GOSUB 2000   '  Compute autoscatic scales.
 1910 GOSUB 2900   '  Define F-keys and  PLOT.
 1920 GOSUB 1800
 1940    IF (FKIN=1) THEN 1045  ' Return to Res. Menu
 1945    IF (FKIN=3) THEN GOSUB 7000 : GOTO 1920  '<<<BSAVE SCREEN BUFFER
 1950    IF (FKIN=10) THEN 8000  '<<< Chain back to FIT SECTION
 1960  BEEP : GOTO 1920
```

```
2000 REM *** AUTOMATIC SCALING   (ARGS :  RESI, RESNAT)
2010 REM  1)  X SCALING :
2020  PLMIN=1E+10 : PLMAI=-PLMIN
2030    FOR I=0 TO NABESRES
2035      FOR J=0 TO NYEARSRES
2040        IF RESI(I,J)<PLMIN THEN PLMIN=RESI(I,J)
2050        IF RESI(I,J)>PLMAI THEN PLMAI=RESI(I,J)
2060    NEXT J,I
2070  RESMAI=PLMAI : RESMIN=PLMIN
2090  PLTNTICS=10
2100 GOSUB 2500
2110  RSIOI=BOTTIC : RSITOPI=TOPTIC
2120  RSIDELTAI=TICDEL : RESIFAC=SCALTIC
2132 REM
2135 REM  2)  Y SCALING :
2140  PLMIN=1E+10 : PLMAI=-PLMIN
2150    FOR I=0 TO NABESRES
2155      FOR J=0 TO NYEARSRES
2160        IF RESNAT(I,J)<PLMIN THEN PLMIN=RESNAT(I,J)
2170        IF RESNAT(I,J)>PLMAI THEN PLMAI=RESNAT(I,J)
2180    NEXT J,I
2190  RESMINY=PLMIN : RESMAIY=PLMAI
2210  RESLABLY$="RESIDUALS"
2220  PLTNTICS=5
2230 GOSUB 2500
2240  RSBOTYI=BOTTIC : RSTOPYI=TOPTIC
2250  RSDELTAYI=TICDEL : RESFACY=SCALTIC
2270 RETURN
2490 REM  ***COMPUTE TIC-MARK VALUES FOR ANY AXIS
2560 REM
2520  SCALTIC=FNSCALE(FNMAI(ABS(PLMIN),PLMAI))
2530  BOTTIC=FNTIC(PLMIN,SCALTIC)
2540  TOPTIC=FNTIC(PLMAI,SCALTIC)
2550  TICDEL=(TOPTIC-BOTTIC)/PLTNTICS
2560    IF (TICDEL)INT(TICDEL)) THEN TICDEL=INT(TICDEL+1)
2570  TOPTIC=BOTTIC+PLTNTICS*TICDEL
2580 RETURN
2900  DFKMAI=3
2910  DFKEY$(1)="F1 : RESIDUALS Menu"
2915  DFKEY$(2)="F3 : BSAVE Screen"
2920  DFKEY$(3)="F10 : ==) FIT Menu"
2940  CLS
2980 REM
2990 REM *** PLOT ROUTINE  : for SCREEN DISPLAY.
3000 REM ***(Inputs : RESI, RESNAT, )
3020  KPOSDEL=80\DFKMAI : KPOSCOL=1
3030    FOR I=1 TO DFKMAI
3040      LOCATE 25,KPOSCOL : PRINT DFKEY$(I);
3050      KPOSCOL=KPOSCOL+KPOSDEL
3060    NEXT
3070    IF BSAVEFLAG1=1 THEN BSAVEFLAG1=0 : RETURN
3080 GOSUB 3400
3090  LOCATE 24,1
3100  PRINT "Y min= ";RESMINY TAB(50) "Y max= ";RESMINY;
3105  LINE (0,191)-(639,191)
3106  RUNNUMPOSI=INSTR(TITPL$," Run #")
3107    IF RUNNUMPOSI<>0 THEN RESTITPL$=LEFT$(TITPL$,RUNNUMPOSI-1) ELSE RESTITPL$=TITPL$
3108  RESTITPL$="Residuals  vs. "+RESILABL$+" - "+RESTITPL$
3110    IF (LEN(RESTITPL$)<64) THEN RESTITPL$=RESTITPL$+"  Run #"+STR$(RUNNUM)
3111    IF LEN(RESTITPL$)<73 THEN 3120
3113  BEEP : LOCATE 23,1
3115  PRINT "***  Graph Title is TOO LONG ...   TRUNCATED ***
3117    FOR I=1 TO 1000 : NEXT
3119  RESTITPL$=LEFT$(RESTITPL$,72)
3120  LPL1=LEN(RESTITPL$)
```

```
3130  LOCATE 1,1 : PRINT TAB(44-LPLI\2) RESTITPL$
3140  LOCATE 23,1 : PRINT SPACE$(79);
3150  LOCATE 23,1 : PRINT TAB(44-LEN(RESXLABL$)\2) RESXLABL$;
3160  PRTICXI=48 : PRTICYI=161
3170 GOSUB 3500                    ' Draw  X - AXIS
3172  PRTICXI=19 : PRTICYI=151 : TICDLYI=1
3173 GOSUB 3700                    ' Draw  Y - AXIS
3180    IF (RESMINY>=0) OR (RESMAXY(=0) THEN 3240
3190      YPL=154-(RSBOTYI*RESCALEY)
3200      LINE (50,YPL)-(639,YPL)        ' Draw  I-AXIS in middle of graph
3240 GOSUB 3900  '  Plot DATA points
3250 RETURN
3395 REM  ***DISPLAY the Y-LABEL.
3400  LPLI=LEN(RESLABLY$)
3405    IF (LPLI)19) THEN 3440
3410  LI=11-LPLI\2
3420    FOR J=1 TO LPLI
3425      LOCATE LI,1 : PRINT MID$(RESLABLY$,J,1)
3426      LI=LI+1
3430    NEXT
3435 RETURN
3440  LOCATE 23,1
3445  PRINT "*** Y-LABEL IS TOO LONG ; truncated to 19 columns ***" : BEEP
3450  RESLABLY$=LEFT$(RESLABLY$,19) : LPLI=19
3455 GOTO 3410
3490 REM  Display the X-axis and tics.
3500  LINE (60,158)-(630,158)
3510  TICVALI=RSIOI
3520    WHILE TICVALI<RSITOPI
3523      XTICI=PRTICXI+12
3526      LINE(XTICI,154)-(XTICI,158)
3530      GOSUB 3630
3540      TICVALI=TICVALI+RSIDELTAI
3550      PRTICXI=PRTICXI+PLTDELXTICI
3560    WEND
3565    IF YEARLYRES THEN 3602
3570  LINE (629,154)-(629,158)   '  Display LAST TIC.
3580  PRTICXI=607 : GOSUB 3630
3590  RESXSCALE=570/(RSITOPI-RSIOI)
3591  MI=FNPLI(RESIMAI*10^RESIFAC)
3597  LINE (MI-2,158)-(MI,152)  ' display a triangle at true DATA maximum
3595  LINE -(MI+2,158)          ' on X axis.
3600 RETURN
3602 REM  ***PRINT 1st and last YEARS as tic values
3604  XTICI=PRTICXI+12
3606  LINE(XTICI,154)-(XTICI,158)
3608  PRTICXI=608 : TICVALI=RSITOPI : GOSUB 3630  ' Last year
3612  RESXSCALE=(NYEARSRES*PLTDELXTICI)/(RSITOPI-RSIOI)
3614 RETURN
3620 REM ***PRINT A TIC  at the proper location .
3630  LOCATE 1,1
3632    IF TICVALI<100 THEN PRINT TICVALI ELSE PRINT MID$(STR$(TICVALI),2)
3640    GET(0,0)-(31,7),NUMIMAGE
3650    PUT(0,0),NUMIMAGE
3660    PUT (PRTICXI,PRTICYI),NUMIMAGE
3670 RETURN
3680 REM
3690 REM  ***Display  Y axis  and tics.
3700  LINE(50,9)-(50,154)    ' Axis at LEFT
3710  LINE(639,154)-(639,9)  ' Axis at  RIGHT.
3720  TICVALI=RSBOTYI
3730    WHILE TICVALI(=RSTOPYI
3740      TYI=PRTICYI+3
3750      LINE (50,TYI)-(54,TYI)
3760      LINE (635,TYI)-(639,TYI)
3770      GOSUB 3630
```

```
3780      TICVALI=TICVALI+RSDELTAYI
3790      PRTICYI=PRTICYI+PLTDELTICYI
3800    WEND
3810  RESCALEY=-140/(RSTOPYI-RSBOTYI)
3820  HY=FNPLY(RESMAXY*10^RESFACY)
3830  LINE (639,HY-2)-(631,HY),TICOLYI
3840  LINE -(639,HY+2),TICOLYI
3850 RETURN
3880 REM
3890 REM  *** PLOT  DATA  POINTS
3900    FOR I=0 TO NAGESRES
3905      FOR J=0 TO NYEARSRES
3910        IPL=FNPLI(RESI(I,J)*10^RESIFAC)
3920        YPL=FNPLY(RESMAT(I,J)*10^RESFACY)
3930        CIRCLE(IPL,YPL),2,,,,.5
3940    NEXT J,I
3950 RETURN
3990 REM  ***PLOT Residuals AGAINST I  (F 1)***
4000    FOR K=0 TO NAGESRES
4010      FOR L=0 TO NYEARSRES
4020        RESI(K,L)=PLOTI0(PLFROW+K,L+1)
4030    NEXT L,K
4040  RESILABL$="I  "
4050 GOTO 1900
4060 REM  ***PLOT Residuals  against YEARS  (F 2)***
4070  RSIOI=FYEARES : RSITOPI=FYEARES+NYEARSRES
4080  PLTDELXTICI=INT(570/NYEARSRES)
4090  RSIDELTAI=1 : RESIMIN=FYEARES : RESIFAC=0
4095  WIDTH "lpt1:",132
4100    FOR I=0 TO NAGESRES
4110      FOR J=0 TO NYEARSRES
4120        RESI(I,J)=J+FYEARES
4130    NEXT J,I
4140  RESILABL$="YEARS" : YEARLYRES=-1
4150 GOSUB 2140  ' Compute Y-scale
4160 GOSUB 2900  ' Define F-keys and plot
4170  PLTDELXTICI=57 : YEARLYRES=0
4180 GOTO 1920
4190 REM  ***PLOT Residuals against OBSERVED Yi  (F 3)***
4200    FOR K=0 TO NAGESRES
4210      FOR L=0 TO NYEARSRES
4220        RESI(K,L)=PLOTY0(PLFROW+K,L+1)
4230    NEXT L,K
4240  RESILABL$="Observed Y"
4250 GOTO 1900
4290 REM  ***PLOT Residuals  against PREDICTED Y  (F 4)***
4300    FOR K=0 TO NAGESRES
4310      FOR L=0 TO NYEARSRES
4320        RESI(K,L)=FNYHAT(PLOTI0(PLFROW+K,L+1))
4330    NEXT L,K
4340  RESILABL$="Predicted Y"
4350 GOTO 1900
5000 REM ***Print residuals on screen***
5010    FOR AGERESI=0 TO NAGESRES
5020      CLS
5030        IF NAGESRES=0 THEN PRINT TAB(35) "RESIDUALS" ELSE PRINT TAB(32) "RESIDUALS - AGE" AGERESI+FAGERES
5034      RUNNUMPOSI=INSTR(TITPL$," Run 0")
5036        IF RUNNUMPOSI<>0 THEN RESTITPL$=LEFT$(TITPL$,RUNNUMPOSI-1) ELSE RESTITPL$=TITPL$
5040        IF (LEN(RESTITPL$)<70) THEN RESTITPL$=RESTITPL$+" - RUN 0"+STR$(RUNNUM)
5050      RESTITPL$=LEFT$(RESTITPL$,79)
5060      LPLI=LEN(RESTITPL$)
5070      PRINT TAB(40-LPLI\2) RESTITPL$ : PRINT
5080      PRINT TAB(9) "Year          I          Observed Y      Predicted Y      Residual"
```

```
5090       FOR YEARRESI=0 TO NYEARSRES
5100         RYEARI=YEARRESI+FYEARES+1900
5110         RI=PLOTI@(PLFROW+AGERESI,YEARRESI+1)
5120         RY=PLOTY@(PLFROW+AGERESI,YEARRESI+1)
5130         RYHAT=FNYHAT(PLOTI@(PLFROW+AGERESI,YEARRESI+1))
5140         PRINT USING "        ####";RYEARI;
5150         PRINT USING "     ##.####^^^^";RI;RY;RYHAT;RESMAT(AGERESI,YEARRESI);
5160         L=CSRLIN : IF L<24 THEN PRINT
5170       NEXT YEARRESI
5180     LOCATE 25,1 : PRINT "Press any key to continue.";
5190     A$=INKEY$ : IF A$="" GOTO 5190
5200   NEXT AGERESI
5210 RETURN
6000 REM ***Print residuals on hardcopy***
6010  WIDTH "lpt1:",132
6020  LPRINT CHR$(15) TAB(35) "RESIDUALS"
6024  RUNNUMPOSI=INSTR(TITPL$," Run #")
6026    IF RUNNUMPOSI<>0 THEN RESTITPL$=LEFT$(TITPL$,RUNNUMPOSI-1) ELSE RESTITPL$=TITPL$
6030  RESTITPL$=RESTITPL$+" - RUN #"+STR$(RUNNUM)
6040  LPLI=LEN(RESTITPL$)
6050    IF LPLI<80 THEN LPRINT TAB(40-LPLI\2) RESTITPL$ ELSE LPRINT RESTITPL$
6060    FOR AGERESI=0 TO NAGESRES
6070      LPRINT : LPRINT
6080        IF NAGESRES<>0 THEN LPRINT TAB(37) "AGE" AGERESI+FAGERES : LPRINT
6090      LPRINT TAB(9) "Year          I          Observed Y      Predicted Y      Residual" : LPRINT
6100        FOR YEARRESI=0 TO NYEARSRES
6110          RYEARI=YEARRESI+FYEARES+1900
6120          RI=PLOTI@(PLFROW+AGERESI,YEARRESI+1)
6130          RY=PLOTY@(PLFROW+AGERESI,YEARRESI+1)
6140          RYHAT=FNYHAT(PLOTI@(PLFROW+AGERESI,YEARRESI+1))
6150          LPRINT USING "        ####";RYEARI;
6160          LPRINT USING "     ##.####^^^^";RI;RY;RYHAT;RESMAT(AGERESI,YEARRESI)
6170    NEXT YEARRESI,AGERESI
6180  WIDTH "lpt1:",80 : LPRINT CHR$(18) : LPRINT : LPRINT
6190 RETURN
7000 REM ***BSAVE the screen buffer
7010  BSAVEFLAGI=1
7020  LOCATE 25,1 : PRINT SPACE$(79);
7030  LINE (0,191)-(639,191),0
7040  LOCATE 22,1 : INPUT "Enter filename:   ",BSAVEFN$
7050  LOCATE 22,1 : PRINT SPACE$(40);
7060  BSAVEFN$="B:"+BSAVEFN$
7070  DEF SEG=&HB800
7080    BSAVE BSAVEFN$,0,&H4000 : DEF SEG
7090 GOSUB 3020
7100  LINE (0,191)-(639,191)
7110 RETURN
8000 REM ***RETURN TO PLOT/FIT PROGRAM (FIT SECTION)***
8005  CLS : PRINT "RETURNING TO FIT PROGRAM..."
8010  ERASE RESI,RESMAT
8015  FROMRESID$=-1
8020 CHAIN "PLOT",1010,ALL
```

APPENDIX 11. Printouts of the example data (from Gray, 1979) used in the cohort analysis program.

A. Input data. Only years 1971 to 1978 and ages 1 to 12 were used in the analysis. The excluded years and ages would not be printed in the run described in the text. The weight at age matrix was expanded from a column vector by the program.

Catch at age : 4VsW Cod - Adjusted Data (RUN 0 1)

YEAR	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978
AGE													
1	2806	548	2495	1426	1293	2311	2383	1418	1482	1792	728	2	177
2	17891	4235	16045	9097	8631	15218	17738	12142	8451	9979	4061	24	153
3	17493	6267	17413	7684	8886	12582	14227	14881	12885	9485	3587	386	1004
4	13973	7989	17783	13724	14802	9146	13361	7507	9947	4341	3713	1073	3450
5	10577	9456	15633	10248	13673	8809	9661	9755	7130	4549	4818	1359	4621
6	4461	4328	8297	6073	4539	10262	8780	3823	2766	2594	2412	871	2441
7	3256	1467	3482	2144	1942	5160	3432	2996	944	2627	1426	501	768
8	1590	1239	895	510	759	1849	1919	3724	1323	612	611	220	213
9	856	664	816	237	236	496	358	1166	413	497	184	128	112
10	496	647	361	50	72	114	393	273	369	660	49	35	80
11	666	325	152	95	137	131	79	299	15	153	22	44	26
12	24	65	211	58	56	72	2	3	5	126	107	55	28
13	14	16	33	12	9	98	37	7	0	36	1	11	26
14	0	5	17	2	12	12	0	5	0	9	4	3	9
15	2	7	1	2	4	51	1	5	0	9	1	2	4
16	1	2	10	2	3	17	1	20	0	18	1	7	2

Weight at age : 4VsW Cod (RUN 0 1)

YEAR	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978
AGE													
1	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
2	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
3	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
4	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
5	1.68	1.68	1.68	1.68	1.68	1.68	1.68	1.68	1.68	1.68	1.68	1.68	1.68
6	2.47	2.47	2.47	2.47	2.47	2.47	2.47	2.47	2.47	2.47	2.47	2.47	2.47
7	3.61	3.61	3.61	3.61	3.61	3.61	3.61	3.61	3.61	3.61	3.61	3.61	3.61
8	5.23	5.23	5.23	5.23	5.23	5.23	5.23	5.23	5.23	5.23	5.23	5.23	5.23
9	5.59	5.59	5.59	5.59	5.59	5.59	5.59	5.59	5.59	5.59	5.59	5.59	5.59
10	6.54	6.54	6.54	6.54	6.54	6.54	6.54	6.54	6.54	6.54	6.54	6.54	6.54
11	7.92	7.92	7.92	7.92	7.92	7.92	7.92	7.92	7.92	7.92	7.92	7.92	7.92
12	9.21	9.21	9.21	9.21	9.21	9.21	9.21	9.21	9.21	9.21	9.21	9.21	9.21
13	10.40	10.40	10.40	10.40	10.40	10.40	10.40	10.40	10.40	10.40	10.40	10.40	10.40
14	9.75	9.75	9.75	9.75	9.75	9.75	9.75	9.75	9.75	9.75	9.75	9.75	9.75
15	8.68	8.68	8.68	8.68	8.68	8.68	8.68	8.68	8.68	8.68	8.68	8.68	8.68
16	12.21	12.21	12.21	12.21	12.21	12.21	12.21	12.21	12.21	12.21	12.21	12.21	12.21

F, last year (PR) : 4VsW Cod (RUN 0 1)

YEAR	1978
AGE	
1	0.0023
2	0.0041
3	0.0180
4	0.0720
5	0.1800
6	0.1800
7	0.1620
8	0.1620
9	0.1620
10	0.1620
11	0.1620
12	0.1620
13	0.1620
14	0.1620
15	0.1620
16	0.1620

F, last age : 4VsW Cod (RUN 0 1)

YEAR	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978
	0.5250	0.5250	0.6150	0.5250	0.7050	0.7650	0.9000	1.0650	1.1850	0.9000	0.8000	0.1350	0.1620

Natural mortality is : .2

Partial recruitment : 4VsW Cod (RUN 0 1)

YEAR	1978
AGE	
1	0.013
2	0.023
3	0.100
4	0.400
5	1.000
6	1.000
7	0.900
8	0.900
9	0.900
10	0.900
11	0.900
12	0.900
13	0.900
14	0.900
15	0.900
16	0.900

Smoothed 4+ RV #'s : 4VsW Cod (RUN 0 1)

YEAR	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978
	0	0	0	0	0	32416	34024	25903	14329	14643	22773	35142	42105

B. Output data as calculated in the tutorial example run. Fishable biomass was calculated from "Midyear biomass" and "Arith. mean PR".

N at beg. year : 4VsW Cod (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
AGE								
1	97691	82236	84234	91859	113601	93447	49905	83552
2	77571	77892	65173	67682	73867	91387	75849	40857
3	64747	49740	47722	42373	47766	51448	71147	62078
4	34315	41626	27850	25607	23033	30525	38876	57901
5	31557	19819	21991	16009	11965	14930	21632	30858
6	26227	17866	7485	9176	6656	5680	7864	16301
7	11910	12187	6683	2669	5012	3102	2468	5650
8	3521	5082	6873	2761	1331	1726	1250	1567
9	1586	1210	2425	2257	1063	536	860	824
10	240	850	666	930	1474	421	272	589
11	149	93	340	299	428	610	300	191
12	147	4	5	8	231	212	479	206

Midyear biomass : 4VsW Cod (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
AGE								
1	17446	14642	15101	16475	20378	16838	9031	15102
2	38512	37801	32581	35252	38221	49963	42544	22873
3	49319	35358	33210	29679	36276	42486	60974	52884
4	32598	37819	26395	21880	23180	32121	43295	63173
5	39891	20462	23868	17342	13849	18204	31547	42871
6	44159	26990	11040	16752	11224	9247	16465	33295
7	28086	33039	15507	6804	10618	7108	7099	17021
8	10793	18359	20599	8961	4417	6374	5367	6839
9	6488	5018	8394	10198	3746	2136	3978	3844
10	979	3515	2917	4124	6202	2325	1493	3212
11	185	171	408	2079	2382	4283	1970	1264
12	834	19	24	36	1227	1182	3735	1583

N at midyear : 4VsW Cod (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
AGE								
1	87232	73209	75506	82373	101891	84189	45155	75512
2	62116	60969	52551	56859	61646	80635	68619	36892
3	51915	37219	34957	31241	38185	44722	64183	55667
4	26079	30256	21116	17504	18544	25697	34636	50539
5	23745	12180	14207	10323	8244	10836	18778	25518
6	17878	10927	4470	6782	4544	3744	6666	13480
7	7780	9152	4296	1885	2941	1969	1967	4715
8	2064	3510	3939	1713	845	1219	1015	1308
9	1161	898	1502	1824	669	382	712	688
10	150	537	446	631	948	355	228	491
11	23	22	51	263	301	541	249	160
12	91	2	3	4	133	128	406	172

Age-specific F : 4VsW Cod (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
AGE								
1	0.0265	0.0325	0.0188	0.0180	0.0176	0.0086	0.0000	0.0023
2	0.2444	0.2899	0.2305	0.1485	0.1617	0.0504	0.0003	0.0041
3	0.2418	0.3800	0.4225	0.4096	0.2478	0.0802	0.0060	0.0180
4	0.3489	0.4381	0.3537	0.5609	0.2336	0.1444	0.0310	0.0720
5	0.3689	0.7738	0.6738	0.6777	0.5450	0.4411	0.0830	0.1800
6	0.5664	0.7833	0.8312	0.4051	0.5634	0.6336	0.1306	0.1800
7	0.6516	0.3728	0.6841	0.4958	0.8659	0.7093	0.2541	0.1620
8	0.8685	0.5401	0.9134	0.7542	0.7096	0.4963	0.2164	0.1620
9	0.4241	0.3962	0.7582	0.2259	0.7268	0.4771	0.1796	0.1620
10	0.7442	0.7159	0.6029	0.5772	0.6827	0.1377	0.1532	0.1620
11	3.5071	2.7343	3.5753	0.0571	0.5034	0.0407	0.1767	0.1620
12	0.7650	0.9000	1.0650	1.1850	0.9000	0.8000	0.1350	0.1620

Total N, beg. year : 4VsW Cod - AGES 4 TO 12 (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
	109652	98737	74318	59718	51192	57742	74002	114087

Total N, midyear : 4VsW Cod - AGES 4 TO 12 (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
	78970	67484	50028	40928	37169	44871	64655	97070

Total B, midyear : 4VsW Cod - AGES 4 TO 12 (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
	164014	145392	109151	88175	76838	82979	114889	173103

Weighted mean F : 4VsW Cod - AGES 5 TO 12 (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
	0.5150	0.6648	0.7607	0.5583	0.6306	0.5052	0.1148	0.1771

Overall F : 4VsW Cod - AGES 5 TO 12 (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
	0.5012	0.6472	0.7427	0.5447	0.6211	0.4938	0.1133	0.0000

Raw PR matrix : 4VsW Cod (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
AGE								
1	0.0076	0.0119	0.0053	0.0152	0.0195	0.0109	0.0002	0.0130
2	0.0697	0.1060	0.0645	0.1253	0.1797	0.0629	0.0014	0.0230
3	0.0689	0.1390	0.1182	0.3456	0.2753	0.1002	0.0237	0.1000
4	0.0995	0.1602	0.0989	0.4733	0.2595	0.1805	0.1219	0.4000
5	0.1052	0.2830	0.1885	0.5719	0.6056	0.5513	0.3267	1.0000
6	0.1615	0.2865	0.2325	0.3418	0.6260	0.7920	0.5139	1.0000
7	0.1858	0.1364	0.1913	0.4184	0.9621	0.8866	1.0000	0.9000
8	0.2476	0.1975	0.2555	0.6364	0.7865	0.6203	0.8516	0.9000
9	0.1209	0.1449	0.2121	0.1906	0.8076	0.5964	0.7070	0.9000
10	0.2122	0.2618	0.1686	0.4871	0.7585	0.1722	0.6030	0.9000
11	1.0000	1.0000	1.0000	0.0482	0.5593	0.0509	0.6954	0.9000
12	0.2181	0.3292	0.2979	1.0000	1.0000	1.0000	0.5313	0.9000

Arith. mean PR : 4VsW Cod - YEARS 1975 TO 1977 (RUN 0 1)

YEAR	1978
AGE	
1	0.0107
2	0.0856
3	0.1401
4	0.1972
5	0.5208
6	0.6782
7	1.0000
8	0.7935
9	0.7410
10	0.5384
11	0.4583
12	0.8866

PR matrix (from AM) : 4VsW Cod (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
AGE								
1	0.0080	0.0125	0.0055	0.0160	0.0206	0.0114	0.0002	0.0137
2	0.0734	0.1117	0.0679	0.1320	0.1892	0.0663	0.0014	0.0242
3	0.0726	0.1467	0.1245	0.3640	0.2899	0.1056	0.0249	0.1053
4	0.1048	0.1687	0.1042	0.4985	0.2733	0.1900	0.1284	0.4212
5	0.1108	0.2980	0.1985	0.6022	0.6378	0.5806	0.3440	1.0531
6	0.1701	0.3017	0.2448	0.3600	0.6592	0.8341	0.5412	1.0531
7	0.1957	0.1436	0.2015	0.4406	1.0132	0.9337	1.0531	0.9478
8	0.2608	0.2080	0.2690	0.6702	0.8303	0.6533	0.8969	0.9476
9	0.1274	0.1526	0.2233	0.2008	0.8505	0.6280	0.7445	0.9478
10	0.2235	0.2757	0.1776	0.5129	0.7988	0.1813	0.6350	0.9478
11	1.0531	1.0531	1.0531	0.0508	0.5890	0.0536	0.7323	0.9478
12	0.2297	0.3466	0.3137	1.0531	1.0531	1.0531	0.5596	0.9478

Fishable biomass : 4VsW Cod (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
AGE								
1	187	157	162	177	218	180	97	162
2	3298	3238	2791	3019	3274	4282	3644	1959
3	6911	4955	4654	4159	5083	5954	8544	7411
4	6430	7460	5206	4316	4572	6336	8540	12461
5	20775	10657	12430	9032	7213	9481	16430	22327
6	29547	18304	7487	11360	7612	6271	11166	22579
7	28086	33039	15507	6804	10618	7108	7099	17021
8	8564	14567	16345	7111	3505	5057	4211	5427
9	4808	3719	6220	7557	2771	1583	2948	2848
10	527	1892	1570	2220	3339	1252	804	1729
11	85	78	187	953	1092	1963	903	579
12	741	17	22	32	1090	1050	3319	1467

Total B, fishable : 4VsW Cod - AGES 1 TO 12 (RUN 0 1)

YEAR	1971	1972	1973	1974	1975	1976	1977	1978
	110359	98082	72580	56738	50386	50516	67703	95910